Promoting Animation Synthesis Through Dual-Channel Fusion

XiaoHong Qiu[1,2], ChaoChao Guo[1(✉)], and Cong Xu[1]

[1] Jiangxi University of Science and Technology, Nanchang 330013, China
1936452531@qq.com

[2] Nanchang Key Laboratory of Virtual Digital Factory and Cultural Communications, Nanchang 330013, People's Republic of China

Abstract. Although animation synthesis technology is widely applied, it also imposes higher demands on the precision of the synthesized animation. This paper employs a more lightweight channel attention module for image feature extraction. Compared to previous channel attention module, this approach utilizes fewer parameters, thereby assisting the network in achieving improved precision. Additionally, it replaces the sigmoid function with the more suitable output function tanh for image generation. Three evaluation metrics show improvements: a 1.3% increase in L1, an 18.9% increase in AED, and a 2.6% increase in AKD. To facilitate better image generation by the generator, improvements are made to the discriminator. Spectral normalization and instance normalization are combined to form a multi-normalization module for normalization during the image down sampling process. Additionally, an adaptive Dual-Channel Fusion output module is employed for the discriminator output, aiding in the rapid convergence of the network. The quality metrics of the generated images demonstrate improvements, with a 4.3% increase in L1, a 23.8% increase in AED, and a 5.5% increase in AKD.

Keywords: Animation Synthesis · Channel Attention · Dual-Channel Fusion

1 Introduction

Animation synthesis finds extensive applications in social platforms, virtual try-on, virtual character generation, and game sprite production [1]. On social platforms, individuals can showcase their positive aspects through videos, but some complex dance moves are challenging for the average person to perform. Traditional methods for animation compositing primarily rely on manually selecting existing actions through video or image editing software for composition. Although this approach can generate new action videos, it is difficult to synthesize previously non-existent body movements, and the synthesis process can be slow. The development of deep learning, convolutional neural networks, and generative adversarial networks have provided effective solutions. The integration of convolutional neural networks and generative adversarial networks in motion transfer has significantly alleviated the aforementioned issues. It not only speeds

C. Wu et al. (Eds.): MONAMI 2023, LNICST 559, pp. 275–285, 2024.
https://doi.org/10.1007/978-3-031-55471-1_21

up the animation compositing process and saves human resources but also generates entirely new actions.

Based on the maturation of generative adversarial network (GAN)-based image synthesis techniques [2], the "Everybody Dance Now" [3] algorithm in 2017 employed a two-stage approach to achieve motion transfer between images and generate animations. The first stage involved extracting key point information from the characters in the images, while the second stage utilized the key point information from the first stage to generate new animations. In 2018, Wiles [4] and Siarohin [5] respectively proposed the single-stage animation synthesis models "x2face" and "Monkey-Net". "x2face" primarily focused on facial expressions of the characters and could not handle other complex movements. "Monkey-Net" employed a self-supervised approach for key point localization and motion information learning but had limited representation capability for intense actions. In 2019, Siarohin [6] introduced the "First Order Motion Model" (FOMM), which used a set of self-learned key points and local affine transformations to describe complex motion processes. Compared to previous methods, this approach not only generated high-quality images and more natural movements but also required less training data, reducing training time and device requirements. However, challenges still exist regarding insufficient feature utilization and the need for further improvement in image quality. The introduction of channel attention, along with new activation functions and network architecture combinations, helps the network capture weighted features and generate higher-quality images.

2 Relate Work

FOMM not only enables the transition of animation compositing technology from two-stage to single-stage but also reduces the requirements for training data, eliminating the need for annotation of the original data. The network model employs the distance between generated images and driving frames as supervision for training. The network can generate new animations based on a static image and a sequence of videos or continuous frames, unlike traditional style transfer techniques, focusing on the motion and pose from the driving video. As shown in Fig. 1, modeling the motion information begins by passing the source image and driving frames through a key point detection network, which employs the SoftMax function to output the same number of key points. The classified key points are used for motion prediction in the motion module. By mapping the key points of the source image to those of the driving frame, the motion trajectories of each classified key point are calculated. Optical flow, a pixel-level technique for predicting the movement trajectory of consecutive frames, is utilized in combination with first-order Taylor expansion for motion modeling. The estimated feature vector containing motion information is then combined with the appearance features of the source image to generate an explicit feature vector through a dense motion field network. For the image generation component, a generator from a generative adversarial network is employed for image synthesis. The first-order motion model can generate new images with different pose, incorporating the motion and pose from the driving frames, based on the style of the source image.

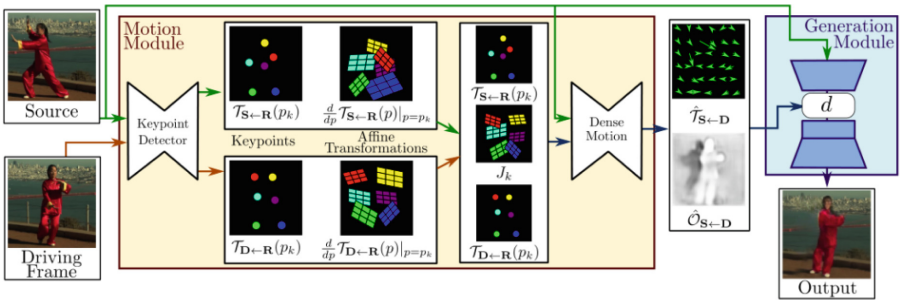

Fig. 1. First-Order Motion Model

The objective function used during the training of a first-order motion model:

$$Loss = L_{perceptual} + L_{gan} + L_{feature\_matching} + L_{equivariance} + L_{equi\,\mathrm{var}\,iance\_value}$$

The objective function is as follows: Where $L_{perceptual}$ represents the perceptual loss of an image. The perceptual loss helps the network learn the contour structure and texture information of the images, serving as the primary loss for training the first-order motion model network. L_{gan} is composed of $L_{generator\_gan}$ and L_{disc_gan} as the losses for the generator and discriminator in a generative adversarial network (GAN). $L_{feature\_matching}$, as a feature matching loss, utilizes the output of the discriminator to minimize the feature distance between the generated image and the real image. $L_{equivariance}$ and $L_{equi\,\mathrm{var}\,iance\_value}$ are losses for the motion pose and real image pose of generating images, obtained by estimating the Jacobian matrix loss of image key points.

3 First-Order Motion Model Animation Synthesis

3.1 Improvement of Hourglass Neural Network in Motion Estimation

The motion estimation in animation generation is divided into two parts, both utilizing a modified hourglass neural network module as the backbone [7]. The process initially involves image feature extraction and classification to estimate the key points of image motion. Sparse matrix of image motion is obtained, which is then used to construct dense optical flow matrix and occlusion mask. The dense optical flow matrix and occlusion mask are fed into the generator, combining the source image information and motion features to generate the image after motion transfer.

In animation synthesis, the Hourglass neural network module is used as the backbone network for motion estimation, as shown in Fig. 2. F stands for feature and cat stands for concatenate. The feature extraction network employs a network structure pattern similar to a feature pyramid, with feature maps of different resolutions. The left side of the network serves as the encoder, gradually reducing the resolution from top to bottom and increasing the number of feature channels from top to bottom. The right side of the network serves as the decoder, with the resolution and number of feature channels changing in the opposite trend. However, each layer with the same resolution is concatenated with the corresponding layer on the left side. After the concatenation is

complete in the decoder layers, the features are fed into convolutions for further feature extraction. The output is then fused with the network information of the same resolution in the next layer, reducing gradient vanishing and allowing for a deeper network design. The channel concatenation not only enhances the model's feature representation capability but also combines surface color information with high-level semantic information, reducing the loss of shallow-level semantics as the network depth increases.

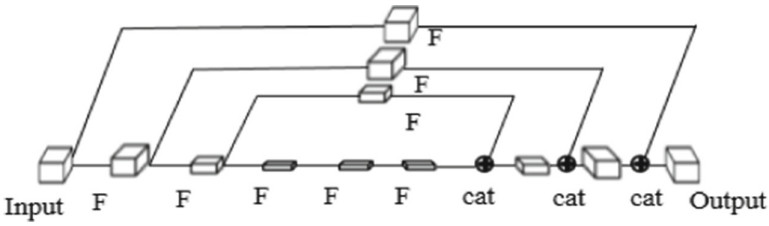

Fig. 2. Hourglass Network Module

Attention modules are often combined with networks [8]. The traditional attention mechanisms in computer vision can be divided into soft [9] and hard attention [10]. Soft attention can update the weight parameters of attention through the forward propagation and gradient backpropagation of the network. Hard attention, on the other hand, emphasizes the importance of specific information and its weight parameters are non-differentiable. This paper utilizes a type of hard attention mechanism, which adjusts the features based on the importance of each channel. This type of attention is known as Channel Attention (CA) [11], and its principle lies in the fact that different channels in a feature map contribute differently to different tasks. Channel attention can be divided into two steps: calculating the importance of each channel and weighting the channels of the input data. In the grid backbone encoding and decoding layers, CA modules containing channel attention are introduced to strengthen the connections between features.

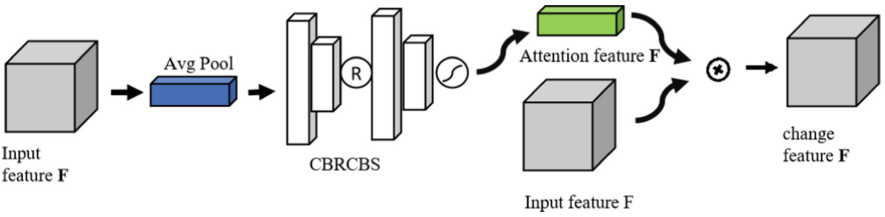

Fig. 3. Channel Attention Module

As shown in Fig. 3, the CA module is first implemented through an adaptive average pooling layer, which adjusts the size of the pooling window based on the size of the input feature map, better preserving the spatial information of the image. When a feature map X is obtained, with dimensions HWC, where H represents height, W represents width, and C represents the number of channels, the output enhancement coefficient is a $1 \times 1 \times$

C vector that represents the importance of each channel. The CBRCRS module uses two convolutional layers to calculate channel features. First, the feature channels are scaled by a factor of a, which requires calculating 1/a times, and then the inverse operation of channel number is performed to align the feature maps, i.e., performing dimensionality increase on the obtained feature vectors. This approach reduces the computational cost by 1/a compared to directly applying two dimension-preserving convolutions on the feature map. Batch normalization is applied during the convolution operation. Additionally, the Relu and Sigmoid functions are used as the activation and output functions, respectively, to obtain weighted feature coefficients. The obtained weighted feature coefficients are multiplied with the original channels to obtain the enhanced part of the feature map. Finally, the enhanced portion of the feature map is added to the original feature map to obtain the enhanced feature map.

3.2 Improved Generative Adversarial Network in FOMM

Estimating the motion between images to obtain dense optical flow images and combining the source images with the motion to generate image animations often requires the use of generative adversarial networks (GANs) [12] to achieve realistic results. GANs serve as framework models capable of generating data samples that follow the probability distribution of the original data.

Tanh-Improved Generator

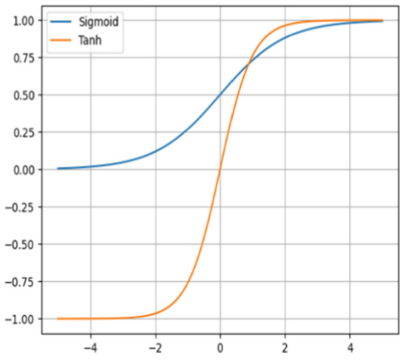

Fig. 4. Sigmoid And Tanh

Choosing an appropriate activation function introduces non-linear transformations to the neural network, thereby facilitating the generator to approximate the real data distribution more closely and enhance its adversarial capability against the discriminator. Different activation functions bring different non-linear transformations to the network. In computer vision tasks, it is necessary to select a suitable activation function based on specific tasks. Images generated using more appropriate activation functions exhibit greater realism and better generalization. Initially, the Sigmoid function was used as the

output function of the network. The characteristics of the network's activation function can be observed from Fig. 4, where the Sigmoid function maps the input of the network layer to a continuous function between 0 and 1, aiding in network convergence. When the Sigmoid function is applied to large or small input parameters, the function's gradient remains relatively smooth. By utilizing steeper gradients, the network can achieve faster convergence, especially when using gradient-based optimization algorithms in the training of deep networks to find more definite descent directions. To address these issues, the Tanh function can be employed in image generation. The Tanh activation function varies between –1 and 1 and has a steep gradient, which helps in the gradient descent during network training.

Multiple Normalization and Adaptive Dual-Channel Fusion
(See Fig. 5).

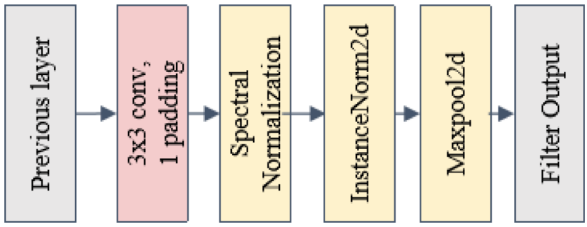

Fig. 5. Multi-normalization Down Sampling

The traditional down sampling module [13] primarily focuses on operations at the feature level, only performing weight normalization on the feature maps in batches or channels [14]. Adding spectral normalization to the down sampling operation, spectral normalization, a weight regularization technique in deep learning. The goal of spectral normalization [15] is to control the range of weights by limiting the spectral norm (maximum singular value) of the weight matrix of a network layer, thereby stabilizing the training process. The main idea behind spectral normalization is to normalize the eigenvectors of the weight matrix during each training iteration, thereby limiting its maximum singular value. This effectively controls the range of weights, preventing them from becoming too large or too small. If spectral normalization alone is used in the discriminator in the network, the generator and discriminator will become more stable, but the overall improvement of the generator by the discriminator will be slower, and the potential of the generator will not be fully stimulated. Instance normalization [16], which includes a scaling factor, can affect spectral normalization but also enhance the training of the discriminator. The combined use of spectral normalization and instance normalization helps control the range of weights, preventing excessive amplification or reduction of weights, and making the network more stable. Instance normalization can reduce internal covariate shift and accelerate the convergence process of the network. Therefore, combining spectral normalization and instance normalization into multi-normalization not only provides stability for the convergence of the adversarial network but also reduces the influence of outliers on the network's trends, helping the generator achieve better convergence and improve the generalization ability of generated images.

It is more important to use a reasonable discriminator output module to obtain the discriminator's discerning output for real and fake sample images. An adaptive dual-channel Fusion discriminator network output module is proposed to assist the discriminator in obtaining a variety of combinations to ensure output stability and accuracy. As shown in the diagram below, different from previous CONCAT and ADD connections for feature maps, two branches are created to expand the network's feature maps. The fused channels of these two branches are jointly transformed, using an adaptive mechanism as the fusion between image features. This enables learning of more reasonable and comprehensive features. When the discriminator assesses the authenticity of an image, it can learn the combination of features through different branches' responses to the feature maps. Finally, by blending the learned weights with the original features, a more accurate output is obtained, enabling targeted pixel-level feature enhancement. This, in turn, helps the discriminator better understand the distribution of image features and further assists the generator in generating more realistic images that conform to the original data distribution (See Fig. 6).

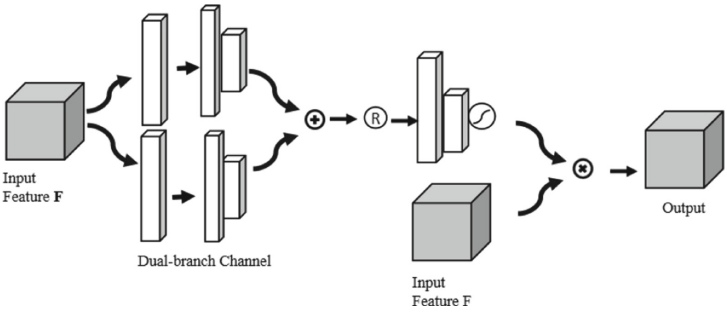

Fig. 6. Adaptive Dual-channel Fusion

Apply a Novel Generative Adversarial Network Function

In general, the loss function of a GAN consists of both the generator loss function and the discriminator loss function, which collectively drive the generator to produce more realistic probabilistic images and the discriminator to better distinguish between real and fake information, strengthening the ongoing adversarial process between the two parties.

$$\mathrm{L}_{gen\_gan} = \frac{1}{KNM} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=1}^{M_k} (1 - D\_G(i,j))^2 \tag{1}$$

$$L_{disc\_gan} = \frac{1}{KNM} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=1}^{M_k} ((1 - D\_R(i,j))^2 + D\_G(i,j)^2) \tag{2}$$

$$L_{gen\_gan} = -\frac{1}{KNM} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=1}^{M_k} (Max(0, \alpha + D\_G(i,j)) \tag{3}$$

$$L_{disc\_gan} = -\frac{1}{KNM}\sum_{k=1}^{K}\sum_{i=1}^{N_k}\sum_{j=1}^{M_k}(Max(0, \beta - D\_R(i,j)) + Max(0, \delta + \mathrm{D\_\ G(i,j)}) \quad (4)$$

In the above formula, L_{gen_gan} represents the generator loss. L_{disc_gan} represents the discriminator loss. K represents the number of images scaled through a Gaussian pyramid, where the dimensions of each feature map are not fixed due to scaling at different ratios. N_k and M_k represent the width and height of each feature map, and α, β, and γ represent modulation factors. The original loss is measured using the squared distance to calculate the distance loss between images. Squared loss has high computational complexity, focuses on fine details of features, and is more susceptible to the impact of noise on the network. The discriminator becomes much stronger in the later stages of training compared to the earlier stages.

4 Experiments

This paper primarily adopts four rigid metrics as accuracy evaluation for the algorithm. Among them, the L1 metric represents the absolute distance between the images generated by the generator and the original real images, disregarding the direction and considering only the magnitude of the distance. AKD represents the average distance of key points between the real and generated images. AED represents the average Euclidean distance between the real and generated images. To ensure the directness of experimental results, EID applies weighted processing to the first three indicators (Figs. 7, 8 and 9, Table 1).

Table 1. Experimental Results Comparison

Methods	L1(‰)	AKD(‰)	AED(‰)	EID(‰)
FOMM	18.26	6.13	4.58	23.62
FOMM + Tanh	18.02	4.97	4.46	22.74
Ours-CL	17.57	4.67	4.33	22.07
Ours	17.47	5.26	4.15	22.18

The above is a comparative experiment conducted on the fashion dataset in the field of computer science. All experiments were conducted without data augmentation, and all data hyperparameters were kept the same, except for the fourth experiment which used a higher number of epochs. The experiments were performed using the same experimental setup, with the A40 graphics card used for training and the 3060 graphics card used for testing. Experiment 1, referred to as FOMM, represents the original author's method and its training results on the dataset. Experiment 2, labeled as FOMM + Tanh, involves modifying the generator's activation function from Sigmoid

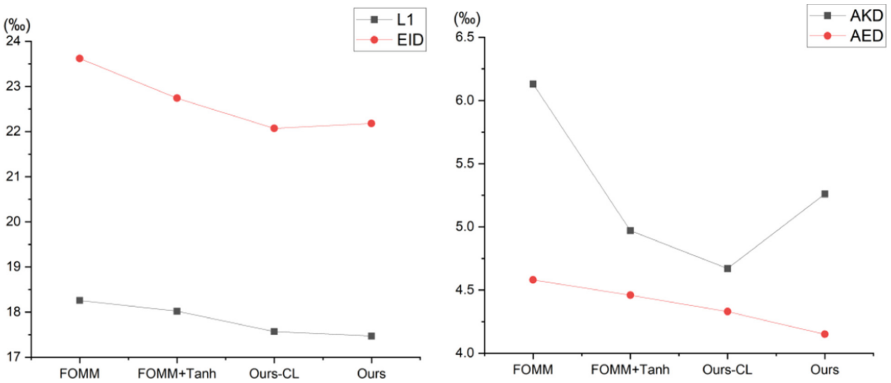

Fig. 7. Experimental Results Comparison

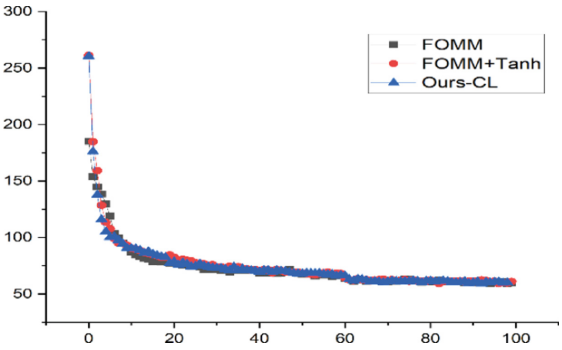

Fig. 8. Perceptual Loss

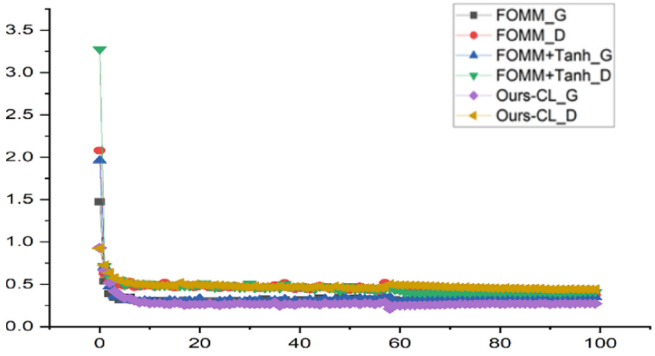

Fig. 9. Gan Loss

to Tanh. Experiment 3, denoted as Ours-CL, combines additional data by incorporating CA and adaptive dual-channel fusion output, using a multi-normalization down sampling

module. The optimized experiment results outperform the previous ones, as evident from the outcomes. The above experiments employed the same constraint function (Fig. 10).

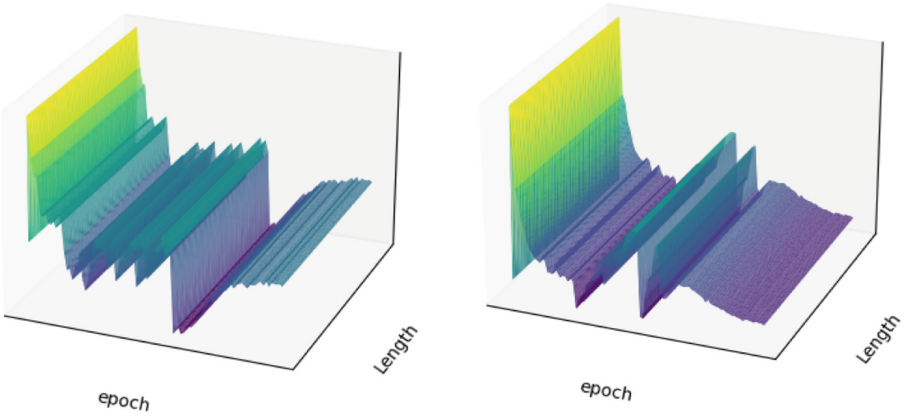

Fig. 10. Feature Matching Loss

The original network convergence process had a common issue, severe oscillation in the later stages of auxiliary loss feature matching. Convergence was modified by introducing a new constraint function, addressing the unstable convergence issue of the auxiliary loss before modification. The network model can converge to a better state, but it requires more time for convergence. The model achieved better results in terms of L1 and AED losses.

5 Conclusion

Although the improved network accuracy has been enhanced, there are still limitations, as the saved model parameters are relatively huge. The proposed new loss function cannot achieve a balance and also increases training time. In future work, reducing the model parameter size can be a key focus, and the loss function needs further improvement to balance the relationship between the original and improved loss functions, thereby promoting the generator to generate more realistic images.

References

1. You, K., Shen, Y., Tao, W.: Research on virtual human value systems based on the value chain theory. Wuhan Univ. J. (Philos. Soc. Sci.) (2023)
2. Isola, P., Zhu, J.Y., Zhou, T., et al.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
3. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: IEEE International Conference on Computer Vision (2019)

4. Wiles, O., Koepke, A., et al.: X2Face: a network for controlling face generation using images, audio, and pose codes. In: Proceedings of the European Conference on Computer Vision (2018)
5. Siarohin, A., Lathuilière, S., Tulyakov, S., et al.: Animating arbitrary objects via deep motion transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2377–2386 (2019)
6. Siarohin, A., Lathuilière, S., Tulyakov, S., et al.: First order motion model for image animation. In: Conference on Neural Information Processing Systems (2019)
7. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
8. Trinh, M.N., Pham, V.T., Tran, T.T.: An attention-PiDi-UNet and focal active contour loss for biomedical image segmentation. In: 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 635–640 (2022)
9. Datta, S.K., Shaikh, M.A., et al.: Soft-Attention Improves Skin Cancer Classification Performance. arXiv preprint arXiv:2105.03358 (2021)
10. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
11. Bastidas, A.A., Tang, H.: Channel attention networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2019)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.: Generative Adversarial Nets. Neural Information Processing Systems. MIT Press, Cambridge (2014)
13. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for Simplicity: The All Convolutional Net. CoRR, abs/1412.6806 (2014)
14. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
15. Miyato, T., Kataoka, T., Koyama, M., et al.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
16. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: the missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)