Optimizing Animation Style Transfer with LS-GAN and GCN-ECANet for Increased Accuracy and Efficiency

Xiaolin Zhao
College of Culture and Media, Yantai Institute of Science and Technology, Yantai, 264000, China
E-mail: zhaouniv25@outlook.com

Keywords: GAN, ECANet, animation style transfer, LS, GCN

Received: July 10, 2025

Animation is a comprehensive art form that integrates painting, film, music, and other artistic styles. As art continues to evolve, animation style transfer has become a key research focus. However, existing transfer methods suffer from low accuracy, poor transfer quality, and long processing times. To address these issues, this study proposes an improved animation style transformation model. This model introduces the least squares method into the generative adversarial network to optimize its loss function, enhance model stability, and improve output quality. Simultaneously combining graph convolutional networks to optimize efficient channel attention networks and enhance their adaptability to complex scenes. The results show that the proposed model achieves an accuracy of 95.1% and a success rate of 86.9% in high-noise environments, with an average accuracy error as low as 0.0076%, significantly outperforming comparison models. Additionally, the model achieves a style transfer similarity of 0.971, a transfer image clarity of 94.8%, and a loss rate of only 0.195%, demonstrating its strong performance. These experimental results indicate that the proposed model exhibits high robustness and transfer stability while efficiently and accurately generating high-quality transferred images. It effectively addresses the issues in traditional animation style transfer models and provides a new method for animation style transfer research.

Povzetek: Raziskava predstavlja izboljšan model za prenos animacijskega sloga z višjo natančnostjo in kakovostjo.

1 Introduction

With the continuous development of society, an increasing number of animation styles and genres have emerged. However, a single artistic style and tone can no longer meet the demand for high-quality artistic expression. The application of intelligent algorithms in this field is expected to bring breakthroughs [1]. Due to the urgent need for high-definition, stable, and diverse animation styles, animation style transfer technology has emerged. This technology transfers the color and style of one image to another while preserving the structure and form of the original image, allowing it to maintain new visual characteristics. As a result, many researchers have explored this technique and widely applied it in advertising, film, and other fields [2]. Existing transfer methods include Convolutional Block Attention Module, Dual Attention Network, and Squeeze-and-Excitation Networks. However, these algorithms suffer from high computational complexity, unstable performance, and low output quality, highlighting the need for a more comprehensive approach [3]. Generative Adversarial Networks (GAN) generate new data similar to real data by training two neural networks adversarially. Through continuous updates and iterations, GAN ultimately produces high-quality images while optimizing its discrimination ability [4]. Efficient Channel Attention Network (ECANet) enhances convolutional layers' ability to extract image features using channel attention modules while maintaining high efficiency. This allows ECANet to effectively perform image classification tasks with strong robustness and generalization capabilities [5]. To improve animation style transfer, this study introduces Least Squares (LS) and Graph Convolutional Networks (GCN) to optimize GAN and ECANet, respectively. Based on the optimized models, a new animation style transfer algorithm is proposed, aiming to efficiently transfer animation images of different styles.

2 Related works

GAN has several advantages, including the ability to generate adversarial samples, strong data generation capability, and effective privacy protection. It is commonly used in image generation, style transfer, and other fields. As a result, many researchers have studied GAN. For instance, to better handle high-quality and diverse time-series data, Brophy E et al. introduced Continuous Variational GAN. By comparing it with traditional GAN models, they evaluated image quality and cross-domain adaptability. The results indicated that this model outperformed traditional GAN in video

processing and realistic image generation [6]. To address the issue of imbalanced transfer data in existing animation style transfer methods, Park C and his team proposed a new GAN-based model. This model generates realistic synthetic data and accurately detects data styles. The results demonstrated that it effectively solved the data imbalance problem and improved the performance of the original system [7]. Zhang J et al. tackled the instability of non-independent and identical data in traditional GAN by introducing a collaborative game parallel learning algorithm. This algorithm enabled cooperative learning between cloud-edge servers and devices, and the results confirmed its effectiveness in resolving the instability issue [8]. ECANet has been widely applied in instance segmentation and image detection, particularly in high-dimensional image processing, as it is less susceptible to environmental interference. For example, to address the weak anti-noise capability of non-local attention, Xia B et al. proposed an efficient non-local contrast attention algorithm based on ECANet. By using a linear function to calculate amplification factors, this method extracted feature information while avoiding noise interference. The results demonstrated that this algorithm could classify images effectively even in high-noise environments [9]. Similarly, to improve the efficiency and accuracy of image semantic segmentation, Guo Z et al. introduced a lightweight segmentation algorithm based on ECANet. By reconstructing spatial and channel information, this algorithm reduced redundant computations and enhanced segmentation precision. Experimental results showed that it significantly improved segmentation efficiency and accuracy [10].

Over time, animation style transfer techniques have become increasingly mature. While researchers continue to explore optimizations, they have also begun to apply these techniques in practical scenarios. For instance, to address the weak control of human animation in existing style transfer methods, Song W et al. proposed a dual-interaction flow fusion method. By integrating motion style features, this approach enhanced the recognition of animated human movements. The study showed that it effectively controlled human animation movements with high precision [11]. Chen M et al. introduced the AnimeGAN model to resolve the issues of feature distortion and texture loss in traditional GAN-based image processing. By incorporating residual networks and normalization, this model improved processing accuracy. Experimental results demonstrated that it not only enhanced texture details but also preserved high-quality image features [12]. Yang S et al. addressed the challenges of misaligned facial features and fixed frame rates in animation video processing by proposing the VToonify framework. By leveraging multi-scale content extraction and a fully convolutional architecture, this framework ensured frame stability and detailed facial feature output. The results confirmed its ability to generate high-quality animation videos [13]. Guo M and his team developed a 3D facial cartoonization tool to tackle the challenge of automatic cartoon face editing. This method utilized adversarial training on a mixed 3D facial dataset to generate high-quality cartoon faces. Experimental results demonstrated that it successfully converted 2D photos into personalized 3D cartoon faces [14]. To address the problem of detail loss and low-quality facial features in image-to-face conversion, Zhang T et al. proposed a GAN-based method incorporating an attention mechanism. By sampling across different spatial and channel dimensions, this approach reduced the complexity of image conversion. The results showed that the converted images retained more complete facial details [15].

In conclusion, while significant progress has been made in animation style transfer research, existing methods still have certain limitations. Under specific conditions and special environments, issues such as local image distortion, long conversion times, and inconsistent animation styles persist. GAN enhances image conversion by making transferred images more similar to the original data, while ECANet maintains computational efficiency while ensuring robustness. Therefore, this study integrates GAN and ECANet to construct a new animation style transfer model, aiming to achieve efficiency and quality image style transfer across different scenarios.

3 Optimization of animation image style transfer based on LS-GAN and GCN-ECANet

3.1 Optimization of GAN structure based on LS method

In animation style conversion, it is necessary to transform an image of one style into another while maintaining the visual quality and details of the image. The generator and discriminator of GAN are both built on deep neural networks, which have high flexibility and powerful feature learning ability. Through adversarial training of the generator and discriminator, GAN can generate high-quality and realistic images. In contrast, some traditional image processing algorithms or simple neural network models may not be able to effectively learn such complex feature mapping relationships. Therefore, this study adopts GAN for animation image style conversion. The generator mainly deceives the discriminator by creating false data, while the discriminator distinguishes between real data and false data provided by the generator and outputs the selected real data to generate new image information [16]. The network structure of GAN is shown in Figure 1.

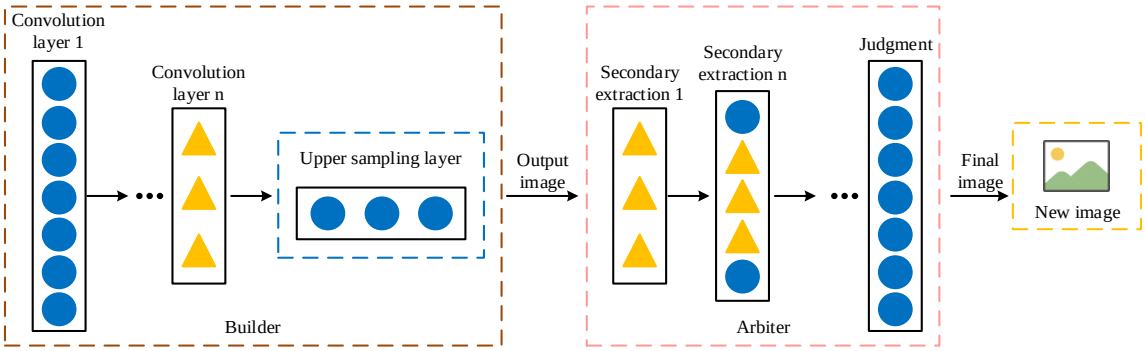


Figure 1: Schematic diagram of GAN architecture

From Figure 1, it can be seen that the generator extracts the original image information through convolutional layers and compresses the image information through different convolutional layers. Then, the upsampling layer restores the feature signals of the optimized original image, bringing them closer to the original image information. Finally, the last convolutional layer of the generator generates fake data of the original image and outputs it to the discriminator. After receiving the fake data, the discriminator performs secondary feature extraction of the image information. Through continuous recognition, the discriminator judges the image information at the last convolutional layer. The image information judged as real is output, completing the image transfer, while the other image information is discarded. The extraction process of the generator is shown in Equation (1).

$$L_D = -\mathrm{IE}_{x\sim Pdata(x)}\left[\log D(x)\right] - \mathrm{IE}_{z\sim P(z)}\left[\log(1-D(G(z)))\right] \quad (1)$$

In Equation (1), L_D represents the loss function of the discriminator, $\mathrm{IE}_{x\sim Pdata(x)}$ represents the expected real sample, $\mathrm{IE}_{z\sim Pz(z)}$ represents the expected generated sample, $D(x)$ and $G(z)$ represent the results of the discriminator and the generator, respectively, $Pdata(x)$ represents the distribution of real data, and $P(z)$ represents the distribution of generated data. The goal of the generator is to deceive the discriminator into outputting its generated data as real data, and its loss function is expressed in Equation (2).

$$L_G = -\mathrm{IE}_{z\sim Pz(z)}\left[\log D(G(z))\right] \quad (2)$$

In Equation (2), L_G represents the loss function of the generator, $\mathrm{IE}_{z\sim Pz(z)}$ represents the expected value of the generated data distribution $P(z)$, $\log D(G(z))$ represents the logarithmic probability of the discriminator discriminating $G(z)$ as real data, and $D(G(z))$ represents the judgment made by the discriminator on the data output by the generator. By these two steps, the extracted original image information is converted into new data for output. The overall expression combining the generator and discriminator is shown in Equation (3).

$$\min_G \max_D V(D,G) = \mathrm{IE}_{x\sim Pdata(x)}\left[\log D(x)\right] + \mathrm{IE}_{z\sim Pz(z)}\left[\log(1-D(G(z)))\right] \quad (3)$$

In Equation (3), $V(D,G)$ represents the objective function of GAN, $\min_G \max_D V(D,G)$ defines the maximum-minimum game process of the GAN, also called the generator-discriminator game. If the fake data provided by the generator does not deceive the discriminator, the generator continuously improves the image parameters to increase the success rate of deceiving the discriminator until it finally deceives it. Meanwhile, the discriminator continuously improves its identification ability to better distinguish fake data until all fake data is identified. In this repeated game process, the result is the generation of samples that are almost indistinguishable from real data. Because a single GAN's training process for image data is not stable enough and the output quality is poor, the study introduces LS to improve it by optimizing the GAN's loss function so that the discriminator can better distinguish between real and fake data. The optimization process is shown in Figure 2.

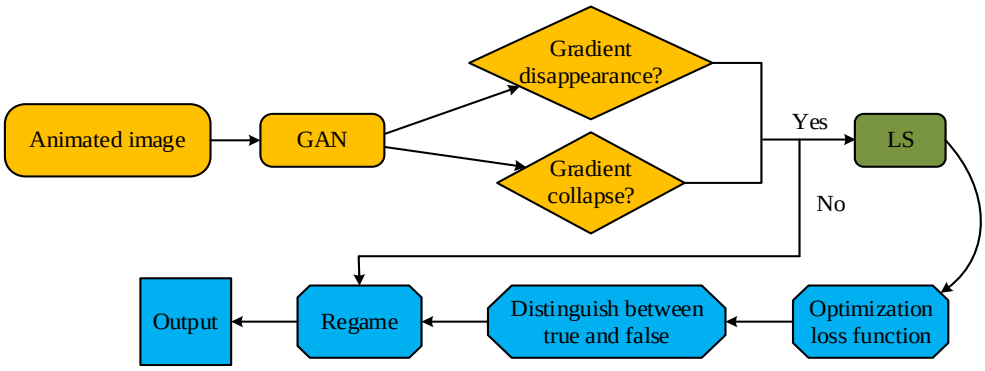


Figure 2: Schematic diagram of LS optimization process for GAN

As shown in Figure 2, traditional GAN uses a cross-entropy loss function, which often causes the gradient of the generator and discriminator to disappear or become unstable during calculation. Therefore, in order to enable the discriminator to better distinguish between real and fake data, LS improves it by optimizing the loss function of GAN. The motivation for introducing the LS loss function is to solve the common problems of gradient vanishing or pattern collapse in original GAN training. The generator loss of the original GAN uses cross entropy, and its gradient tends to saturate and disappear when the discriminator is too confident, hindering the generator's update. The LS loss sets a specific target label for the output of the discriminator, providing a smoother and more stable gradient signal by minimizing the mean square error (MSE) between the discriminator's predicted value and the target label. For the generator, its new loss function is expressed in Equation (4).

$$L_G = \frac{1}{2}\mathrm{IE}_{n\sim Pn}\left[(G(n))-1)^2\right] \tag{4}$$

In Equation (4), $\mathrm{IE}_{n\sim Pn}$ represents the expected value of the generated data distribution $P(n)$, $G(n)$ defines the process in which the generator converts the extracted image information into fake data. For the discriminator, its new loss function is expressed in Equation (5).

$$L_D = \frac{1}{2}\mathrm{IE}_{m\sim Pdata(m)}\left[(D(m)-1)^2\right] + \frac{1}{2}\mathrm{IE}_{n\sim Pn}\left[(D(G(n))-0)^2\right] \tag{5}$$

In Equation (5), $D(m)$ defines the judgment result of the discriminator for the real data m, and $(D(G(n))$ represents the judgment result of the discriminator for the fake data output by the generator. The goal of the discriminator is to minimize this loss function to accurately distinguish between real data and fake data. Through this optimization, LS-GAN can expand the distribution area between real and fake data, thereby improving the recognition ability of the discriminator and enabling the generator to produce higher quality images, rather than focusing solely on whether the discriminator can separate the two as in traditional GANs [17]. The process of animation style transfer using LS-GAN is shown in Figure 3.

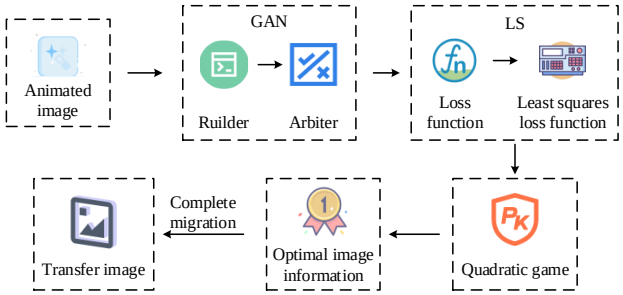


Figure 3: Animation style transfer process based on LS-GAN

As shown in Figure 3, LS-GAN first preprocesses the image set with different animation styles, then initializes the parameters of the generator and discriminator. The generator generates simulated images similar to the original images and then optimizes the simulated images using the least-squares loss function to output the best simulated image. Afterward, the discriminator checks and judges the image. If it successfully deceives the discriminator, it is output, if it fails, it is discarded. The generator then continues to optimize the simulated image using the least-squares loss function until it successfully deceives the discriminator. Through continuous iterative updates in the repeated game process, a simulated image with almost the same style as the original image is generated, completing the animation style transfer.

3.2 Construction of animation style transfer model based on GCN-ECANet and LS-GAN

Although LS-GAN demonstrates strong performance in animation style transfer, it still faces issues like high computational cost, low efficiency, and vulnerability to collapse. In the transition of animation style, the channel information of the image is crucial for the expression of style features. ECANet introduces a channel attention mechanism for one-dimensional convolution operations, which can obtain long-term dependencies between channels at a lower computational cost. Compared with traditional self attention mechanisms, ECANet's one-dimensional convolution operation is more efficient,

reducing computational complexity and parameter count while ensuring performance, making it more suitable for processing large-scale animated image data [18]. Therefore, the study uses ECANet to compensate for the shortcomings of LS-GAN in animation style transfer. Its specific structure is shown in Figure 4.

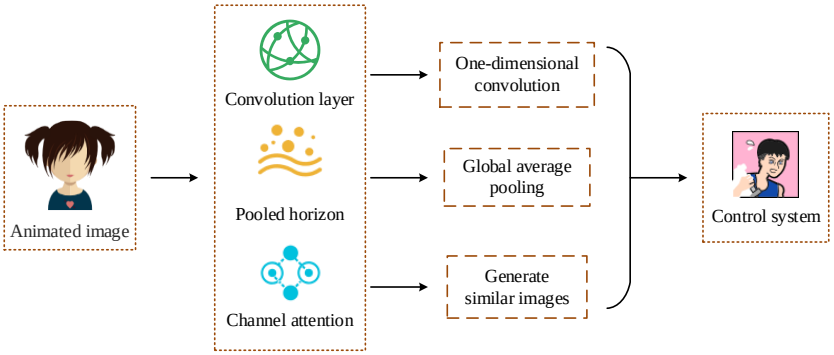


Figure 4: Schematic diagram of ECANet structure

From Figure 4, it can be seen that ECANet mainly consists of convolutional neural networks, pooling layers, and channel attention modules. It uses a standard convolutional neural network to extract features from the image signal, then performs global average pooling in the pooling layer to generate a vector with consistent channel numbers. Finally, the channel attention module learns from the vector and generates new image information similar to the original image's style [19]. The basic one-dimensional convolution process is shown in Equation (6).

$$z_c = \sum_{m=-\frac{k-1}{2}}^{\frac{k-1}{2}} W_m \cdot y(c+m) \bmod C \tag{6}$$

In Equation (6), z_c represents the output of the c-th channel after one-dimensional convolution operation, m represents the movement step size of the convolution kernel in the channel dimension, C represents the total number of channels, $y(c+m) \bmod C$ ensures that the operation is performed within the channel number range, k represents the one-dimensional convolution operation, y represents the channels, and W_m is the weight of the one-dimensional convolution operation. This convolution operation captures the dependencies between different channels in space, thereby obtaining image information with different features and outputting it to the next operation module. The expression for global average pooling is shown in Equation (7).

$$y_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{c,i,j} \quad c = 1, 2, \cdots, C \tag{7}$$

In Equation (7), y_c represents the c-th element in channel y, $X_{c,i,j}$ represents the feature style information X at the c-th channel, i-th row, and j-th column, while H and W are the height and width of the feature image style information. Global average pooling processes the feature image information in the spatial dimension, averaging the image information in each channel and outputting the result, as shown in Equation (8).

$$X'_{c,i,j} = \omega_c \cdot X_{c,i,j} \quad c = 1, 2, \cdots, C; \quad i = 1, \cdots, H; \quad j = 1, \cdots, W \tag{8}$$

In Equation (8), $X'_{c,i,j}$ represents the feature map output value weighted by the channel attention module, ω_c represents the attention weight of each channel. By performing these three operations, different styles of animation image transfer can be achieved. ECANet. However, ECANet mainly relies on channel attention and has limited modeling ability for spatial relationships, which limits its performance in animation scenes with complex spatial structures or character pose changes. GCN excels at handling graph structured data and can explicitly model the spatial relationships between nodes. Therefore, in order to enhance the model's perception of the spatial structure of animated images, this study introduces a GCN layer to preprocess the input features of ECANet. The optimization process is shown in Figure 5.

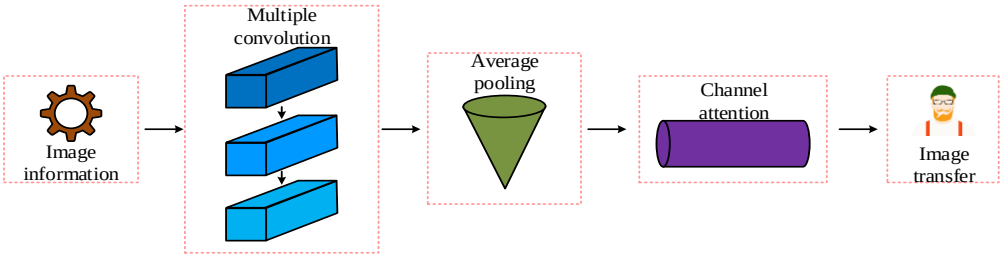


Figure 5: Schematic diagram of GCN's optimization process for ECANet

As shown in Figure 5, traditional convolution operations perform weighted summation on images in a local spatial region, while GCN stacks convolution operations to form a multi-layer convolutional network. The output image information from the previous layer is used as input for the next layer's convolution operation. By extracting features through multiple layers of convolution, optimal image information is gradually obtained [20]. The optimized image style information is then passed into the ECANet operation unit for global average pooling, and the results are output by the channel attention module. The expression for multi-layer convolution operations is shown in Equation (9).

$$H^{(l+1)} = \sigma(\hat{D}-\frac{1}{2}\hat{A}\hat{D}-\frac{1}{2}H^{(l)}W^{(l)}) \quad (9)$$

In Equation (9), $H^{(l+1)}$ represents the feature matrix of the $l+1$-th layer node, σ is the activation function, $\hat{A}$ is the adjacency matrix plus the identity matrix, $\hat{D}$ defines the degree matrix of $\hat{A}$, $H^{(l)}$ represents the feature matrix of the l-th layer node, and $W^{(l)}$ is the weight matrix for the l-th layer. The optimized image style information is then input into the pooling layer of ECANet for global average pooling, as shown in Equation (10).

$$v_p = \frac{1}{P\times H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} O_{c,i,j} \quad p=1,2,\cdots,P \quad (10)$$

In Equation (10), v_p represents the image information output after GCN processing, $P\times H\times W$ represents the size of the input image, and $O_{c,i,j}$ represents the p-th channel element in the i-th row and j-th column of the input image information. The size of each convolution kernel is adaptively determined, and attention weights are generated, as shown in Equation (11).

$$\omega_p = \sigma(t_p) = \frac{1}{1+e^{-t_p}} \quad p=1,\cdots,P \quad (11)$$

In Equation (11), ω_p represents the attention weight of the p-th channel, t_p represents the input value of the p-th channel, $\sigma(t_p)$ represents the output of the Sigmoid activation function. The weight is then input into the feature image information for parameter adjustment and optimization. After processing by the channel attention module, the new feature image is output, as shown in Equation (12).

$$Z = \text{Softmax}(W_{out}X'_{ECANet}+b_{out}) \quad (12)$$

In Equation (12), Z represents the output of the Softmax function, X'_{ECANet} represents the feature input from ECANet, W_{out} defines the weight matrix of the final output layer, and b_{out} is its bias vector. GCN-ECANet not only efficiently and accurately performs animation image style transfer but also exhibits good robustness, avoiding interference from external environments. Therefore, the study constructs an animation style transfer model based on LS-GAN and GCN-ECANet, as shown in Figure 6.

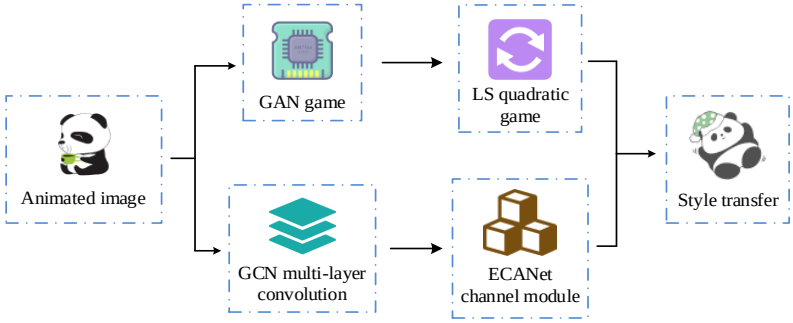


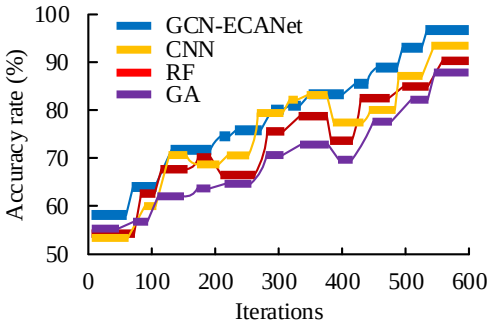
Figure 6: Flowchart of animation image style transfer model

From Figure 6, it can be seen that this model simultaneously activates LS-GAN and GCN-ECANet when receiving image information. From Figure 6, it can be seen that the model simultaneously activates LS-GAN and GCN ECANet when receiving image information. Specifically, the generator of LS-GAN extracts the original image information through convolutional layers, and restores the optimized image features through upsampling layers to generate a simulated image similar to the original image. The discriminator is responsible for judging the generated image. By continuously interacting with the generator and discriminator, new feature images that are closest to the true image style information are generated and output to the LS module to optimize and filter the image style information. The GCN module in GCN ECANet extracts image features through multi-layer convolution, gradually obtaining the best image information. The optimized image style information is passed to the pooling layer of ECANet for global average pooling. The channel attention module generates attention weights and adjusts parameters to output the final image result. If the LS-GAN encounters image style collapse or the image style information is forgotten by the discriminator, the GCN-ECANet receives it for secondary image generation, effectively solving the collapse issue in traditional GANs during animation style transfer.

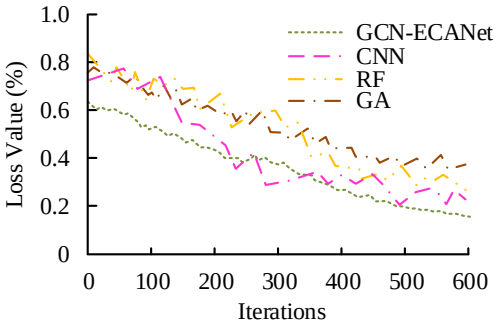
4 Performance validation of animation style transfer model

4.1 Performance evaluation of GCN-ECANet for complex animation images

To verify the performance of the GCN-ECANet algorithm, the study compared it with Convolutional Neural Network (CNN), Random Forest (RF), and Genetic Algorithm (GA) through a series of experiments. The experimental parameters were as follows: the operating system was Windows 11 Professional, the CPU was Intel Core i7-13600F, the GPU was NVIDIA RTX 3090, the programming language was Python 3.8, the memory was 64GB DDR4, 3600MHz, and the storage was a 2TB SSD with an additional 1TB HDD. The initial learning rate was set to 0.001, the batch size was set to 32, the weight of style loss was set to 10, and the weight of content loss was set to 1. The model was trained for 600 epochs, each containing 100 iterations, using the Adam optimizer. During the training process, an early stopping mechanism was used to prevent overfitting when the loss on the validation set did not significantly decrease for 10 consecutive epochs. A video frame from a randomly selected anime-style film in the AnimeGAN Dataset was used to generate an image. Divide the dataset into training, validation, and testing sets in a ratio of 7:2:1. The study compared the accuracy, loss, image transfer success rate in a noisy environment, and image clarity of the GCN-ECANet algorithm against the three other algorithms. The results of the animation style transfer accuracy and loss comparison are shown in Figure 7.



(a) Comparison results of animation image transfer accuracy



(a) Results of comparison of loss rate of animation image migration

Figure 7: Comparison of accuracy and loss rate of each algorithm

As seen in Figure 7(a), when the number of iterations reached 590, the accuracy of GCN-ECANet peaked at 95.1%, which was significantly higher than CNN's 91.3%, RF's 88.9%, and GA's 87.5%. The accuracy of the proposed algorithm increased steadily with the number of iterations, approximately following a linear trend. This result validates the effectiveness of introducing GCN to optimize ECANet. GCN models the spatial dependency relationships between feature points in animated images through multi-layer graph convolution operations, significantly improving the model's understanding of complex animated scene structures. Meanwhile, ECANet can precisely focus on key channel features related to style. The combination of the two enables the model to more accurately capture and transfer the essential features of animation styles, thereby achieving higher accuracy. From Figure 7(b), it can be observed that the loss rate of GCN-ECANet stabilized around 570 iterations, with a minimum of 0.188% and an average loss rate of 0.327%, all of which were lower than those of the comparison algorithms. The stable low loss rate indicates that the GCN-ECANet model has good convergence and robustness during the training process. This is mainly due to the structured spatial information provided by GCN enhancing the consistency of feature representation, as well as the effective suppression of noise and irrelevant information by ECANet, reducing uncertainty in the model optimization process. The study then compared the image transfer success rate of each algorithm in a high-noise environment, as shown in Figure 8.

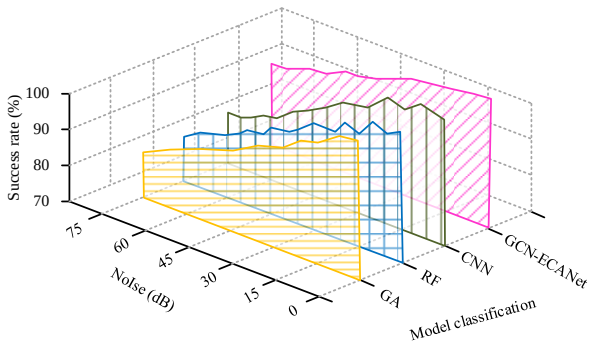
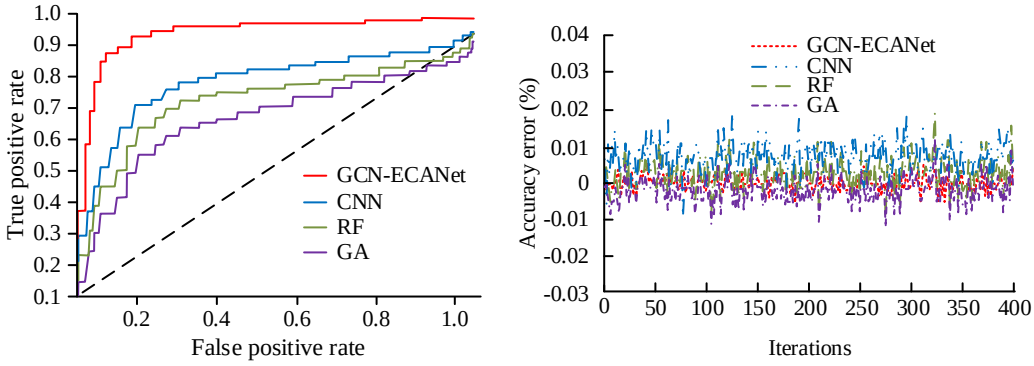


Figure 8: Transfer success rate results under strong noise

As shown in Figure 8, the success rate of GCN-ECANet under 75db noise remained high at 86.9%, far surpassing CNN's 76.4%, RF's 74.8%, and GA's 72.9%. As the noise level increased, the success rate of the proposed algorithm showed little change. Although there was an overall decline, the success rate decreased by only 10.4% from 97.3% at 0db, which was much less than the decline observed in the three comparison algorithms. This also suggests that under extreme noise

conditions, the extraction of key spatial structures and channel features by the model may still be affected. This interference may originate from the complete destruction of local key feature points by noise, making it difficult for GCN to establish effective spatial dependencies, or causing ECANet to have difficulty accurately focusing. For RF, the success rate dropped from 89.6% to 74.8% at 75db noise, with significant fluctuations during the process, showing instability compared to the proposed algorithm. The results show that even in a 75dB strong noise environment, the proposed GCN-ECANet model still maintains excellent transmission success rate, confirming its anti-interference performance. This is because GCN's ability to extract global structural information makes it insensitive to local noise disturbances and able to maintain a grasp of the overall structure of the image. Meanwhile, ECANet can adaptively weight important channels and effectively filter out irrelevant feature interference introduced by noise. Finally, the accuracy error and the Area Under Curve (AUC) of the four algorithms were compared, with the results shown in Figure 9.



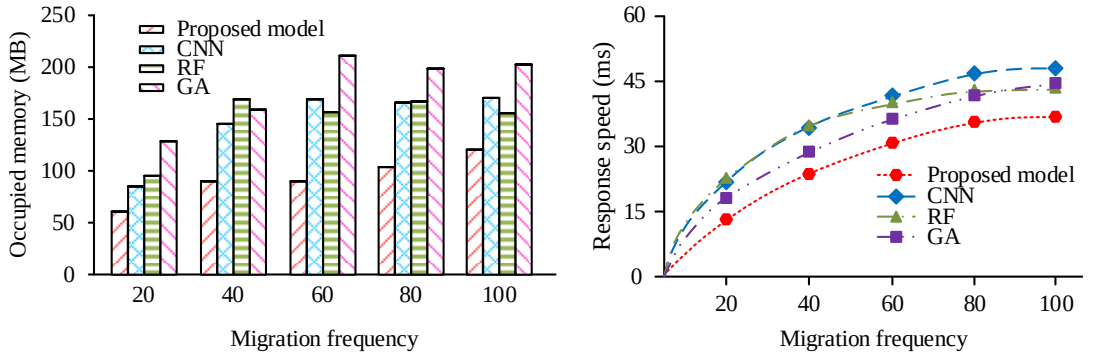
(a) ROC curve comparison among algorithms (b) Comparison of accuracy and error of different algorithms

Figure 9: Comparison of accuracy, error and AUC values

As seen in Figure 9(a), the AUC of GCN-ECANet's Receiver Operating Characteristic (ROC) curve stabilized at a false positive rate of 0.27, reaching a maximum value of 0.994 at 0.88, which was the closest to 1, significantly outperforming CNN (0.908), RF (0.897), and GA (0.874). From Figure 9(b), it is evident that the accuracy error of GCN-ECANet fluctuated the least, with an average error of 0.0076%, which was much lower than CNN's 0.016%, RF's 0.013%, and GA's 0.012%. The accuracy error fluctuation of the proposed algorithm remained smooth, without the instability seen in the ROC curves of CNN and RF. In conclusion, the proposed algorithm demonstrated high accuracy, low error, and resistance to strong noise interference, maintaining over 86% success rate in high-noise environments, fully reflecting its superiority.

4.2 Practical application of transfer model based on LS-GAN and GCN-ECANet

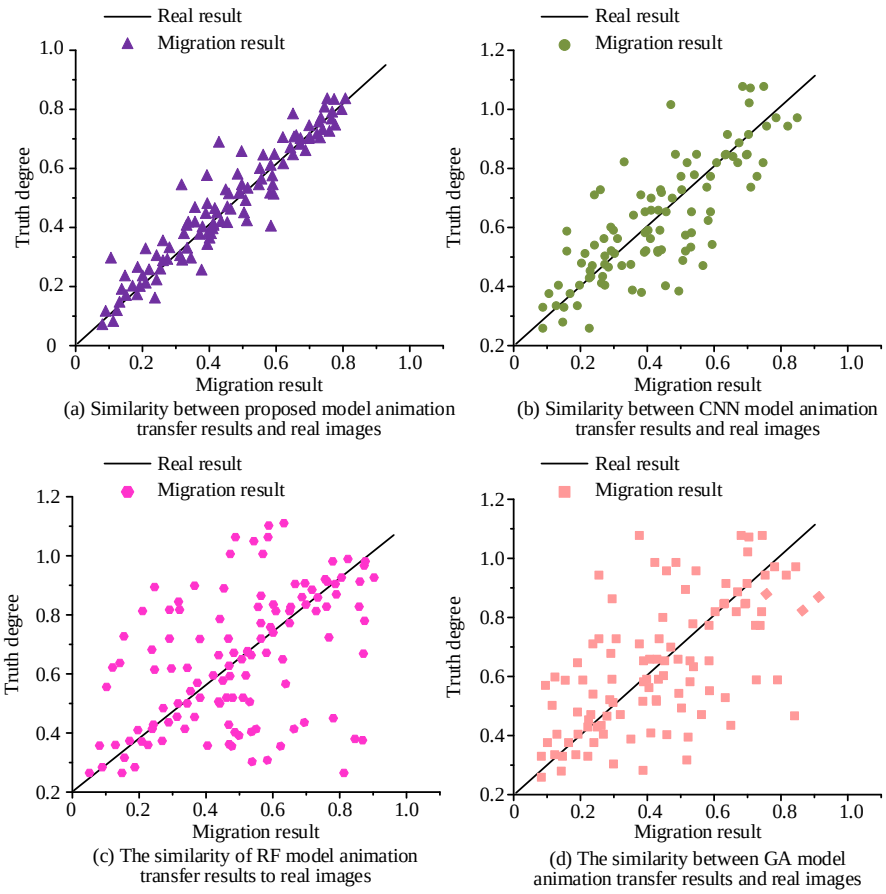
In the previous section, the powerful performance of the core component GCN-ECANet in complex noise environments was verified. Based on this, this section aims to evaluate the comprehensive performance of the complete integrated model (LS-GAN+GCN-ECANet) in practical application scenarios, with a focus on its efficiency and generation quality. The study used a classic animation, "SpongeBob SquarePants", and performed style transfer on a specific frame of the protagonist SpongeBob from a video clip. All models are tested on the same hardware devices, including the same CPU, GPU, memory, and other configurations, to ensure the objectivity and comparability of the test results. In order to reduce the randomness and randomness of the experimental results, 10 repeated experiments were conducted on the performance evaluation indicators of each model, and the average was taken as the final evaluation result. The proposed model was compared with CNN, RF, and GA models. First, the response speed and memory usage of each model in transferring the SpongeBob image were compared, with the results shown in Figure 10.



(a) Comparison of response speed of each model (b) Comparison of response speed of each model

Figure 10: Comparison of image migration time and memory usage

As shown in Figure 10(a), the memory usage of the proposed model increased smoothly without any sudden large spikes. Its maximum memory usage was only 118MB, significantly lower than CNN's 164MB, RF's 166MB, and GA's 217MB. Meanwhile, the average and minimum memory usage of the proposed model were 97MB and 57MB, respectively, both better than the comparison models. Figure 10(b) shows that the response speeds of the four models gradually stabilized after about 80 transfers, with the proposed model having a maximum response speed of only 34ms, significantly lower than CNN's 48ms, RF's 37ms, and GA's 39ms. As the number of transfers increased, the response speed of the proposed model remained lower than the comparison models, demonstrating its strong computational power. The results show that the proposed model exhibits low memory usage and high response speed. The high efficiency of ECANet itself is the foundation, and GAN uses LS loss instead of cross entropy to provide smoother gradients, avoiding the additional computational overhead caused by oscillations and crashes commonly seen in traditional GAN training, thus demonstrating high efficiency in inference. Next, the structural similarity of the transferred images was compared, with the results shown in Figure 11.



(a) Similarity between proposed model animation transfer results and real images

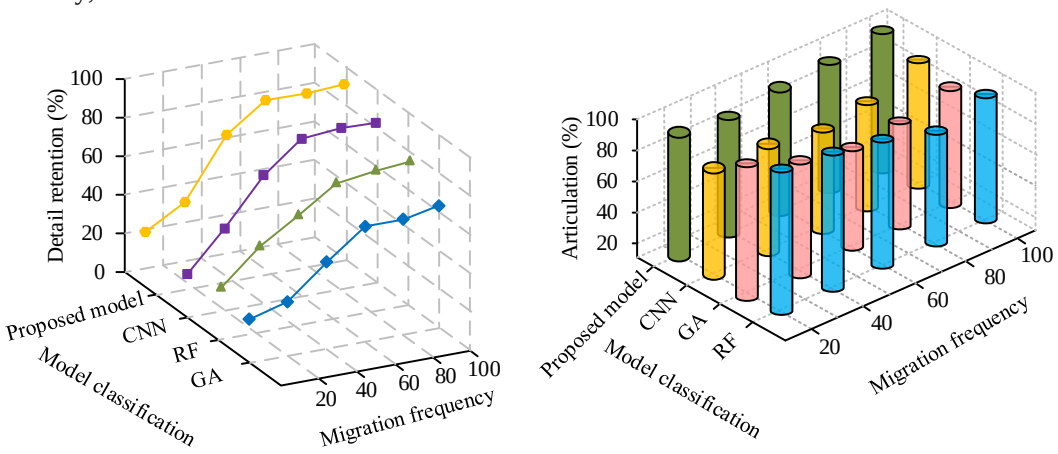
(b) Similarity between CNN model animation transfer results and real images

(c) The similarity of RF model animation transfer results to real images

(d) The similarity between GA model animation transfer results and real images

Figure 11: Comparison of animation migration structure similarity

As shown in Figure 11(a), the structural similarity between the transferred image and the real image was 0.971 for the proposed model, which was closest to 1, significantly higher than CNN's 0.843, RF's 0.519, and GA's 0.548. This indicated that the proposed model's animation style transfer was the closest to the original image. Additionally, the RF model's transfer results were highly dispersed, with one sample having a similarity of only 0.137, further reflecting the accuracy of the proposed model in animation style transfer. Finally, the image clarity and detail retention of the transferred images were compared, with the results shown in Figure 12.



(a) Comparison of detail retention of each model (b) Comparison of the definition of each model

Figure 12: Comparison of image migration clarity and detail retention

As seen in Figure 12(a), the proposed model retained 96.3% of the image details, significantly higher than CNN's 82.1%, RF's 78.6%, and GA's 62.7%, greatly preserving the style features of the original animation image. Figure 12(b) shows that as the number of transfers continued to increase, the image clarity of the animation style transfer remained nearly unchanged, with a slight improvement. When the number of transfers reached 100, the clarity of the proposed model was 94.8%, significantly higher than CNN's 83.6%, RF's 84.4%, and GA's 79.3%. The extremely high detail retention rate and image clarity are direct evidence that the proposed model can generate high-quality images. This is mainly due to the integration framework combining the stable high-quality generation capability of LS-GAN and the advantages of GCN ECANet in spatial structure preservation and detail enhancement, which can accurately strengthen feature channels related to style texture details and focus on generating high-definition images with rich details. To further validate the superiority of the proposed model, it was compared with three mature image style conversion algorithms, CycleGAN, MoCoGAN, and Faster CNN. This study used the AnimeGAN dataset and added 60dB Gaussian noise to the test set to simulate practical application scenarios. Structural similarity (SSIM), Fr é chet Inception Distance (FID), Learned Perceptual Image Patch Similarity (LPIPS), and Time Cost are used as performance evaluation metrics. The performance test results of the four style conversion methods are shown in Table 1.

Table 1: Performance test results of four style conversion methods

Method	SSIM	FID	LPIPS	Time Cost/ms
CycleGAN	0.84	42.67	0.38	52.15
MoCoGAN	0.83	38.88	0.41	68.22
Faster-CNN	0.86	51.32	0.30	28.07
Proposed model	0.93	21.51	0.62	33.86

From Table 1, it can be seen that the animation style transformation model proposed by combining LS-GAN and GCN-ECANet performs better than existing methods in terms of SSIM, FID, and LPIPS, with scores of 0.93, 21.51, and 0.62, respectively. Although slightly more time-consuming than the Faster CNN algorithm, it is lower than CycleGAN and MoCoGAN, with the best overall performance.

5 Conclusion

To address the issues of low accuracy, efficiency, and poor image quality in existing animation style transfer models, the study introduced LS optimization to traditional GAN, then incorporated GCN to improve ECANet. A new animation style transfer model was constructed by combining LS-GAN and GCN-ECANet. This model compensated for the drawbacks of GAN and ECANet through LS and GCN, while utilizing their advantages to achieve efficient, accurate, and high-quality animation style image transfer.

Experiments were conducted to compare various performance metrics of the model, including accuracy and error, loss rate, transfer success rate under strong noise, area under the ROC curve, image style transfer response speed and memory usage, structure similarity, and clarity of the transferred images. The results showed that the proposed model achieved an accuracy of 95.1%, with an average error of only 0.0076%. Under a strong noise environment of 75 dB, the transfer success rate was 86.9%, significantly outperforming the comparison models, demonstrating its ability to generate high-quality images. As the transfer iterations accumulated, the model's maximum response speed and memory usage were 34 ms and 118 MB, respectively, with a loss rate of only 0.195%. The structure similarity between the transferred image and the original reached 0.971, and the maximum clarity was 94.8%, all of which were significantly better than those of the three comparison models. This highlights the high stability and good robustness of the proposed model. In conclusion, the research model fully meets the current needs for animation style transfer, ensuring high accuracy and success rates while maintaining image quality comparable to the original. Although the performance of the model was already excellent, the experiments did not focus on animation style classification, only conducting comparisons with cartoon styles. Therefore, further research should focus on this aspect in future studies.

References

[1] Nir O, Rapoport G, Shamir A. CAST: Character labeling in Animation using Self-supervision by Tracking. Computer Graphics Forum. 2022, 41(2): 135-145. https://doi.org/10.1111/cgf.14464

[2] Li H, Zhu W. Art image style conversion based on multi-scale feature fusion network. Informatica, 2024, 48(10). https://doi.org/10.31449/inf.v48i10.5960

[3] Cheng J, Yang L, Tong S. Painting style and sentiment recognition using multi-feature fusion and style migration techniques. Informatica, 2024, 48(21): 127-138. https://doi.org/10.31449/inf.v48i21.6891

[4] Kozak J P, Zhang R, Porter M, Song Q, Liu J, Wang B, Zhang Y. Stability, reliability, and robustness of GaN power devices: A review. IEEE Transactions on Power Electronics, 2023, 38(7): 8442-8471. https://doi.org/10.1109/TPEL.2023.3266365

[5] Shaik N S, Cherukuri T K. Multi-level attention network: application to brain tumor classification. Signal, Image and Video Processing, 2022, 16(3): 817-824. https://doi.org/10.1007/s11760-021-02022-0

[6] Brophy E, Wang Z, She Q, Ward T. Generative adversarial networks in time series: A systematic literature review. ACM Computing Surveys, 2023, 55(10): 1-31. https://doi.org/10.1145/3559540

[7] Park C, Lee J, Kim Y, Park J G, Kim H, Hong D. An enhanced AI-based network intrusion detection system using generative adversarial networks. IEEE Internet of Things Journal, 2022, 10(3): 2330-2345. https://doi.org/10.1109/JIOT.2022.3211346

[8] Zhang J, Zhao L, Yu K, Min G, Al-Dubai A Y. Zomaya A Y. A novel federated learning scheme for generative adversarial networks. IEEE Transactions on Mobile Computing, 2023, 23(5): 3633-3649. https://doi.org/10.1109/TMC.2023.3278668

[9] Xia B, Hang Y, Tian Y, Yang W, Liao Q, Zhou J. Efficient non-local contrastive attention for image super-resolution. Proceedings of the AAAI conference on artificial intelligence. 2022, 36(3): 2759-2767. https://doi.org/10.1609/aaai.v36i3.20179

[10] Guo Z, Ma D, Luo X. A lightweight semantic segmentation algorithm integrating CA and ECA-Net modules. Optoelectronics Letters, 2024, 20(9): 568-576. https://doi.org/10.1007/s11801-024-3241-z

[11] Song W, Jin X, Li S, Chen, C, Hao A, Hou X. Finestyle: Semantic-aware fine-grained motion style transfer with dual interactive-flow fusion. IEEE Transactions on Visualization and Computer Graphics, 2023, 29(11): 4361-4371. https://doi.org/10.1109/TVCG.2023.3320216

[12] Chen M, Dai H, Wei S, Hu Z. Linear-ResNet GAN-based anime style transfer of face images. Signal, Image and Video Processing, 2023, 17(6): 3237-3245. https://doi.org/10.1007/s11760-023-02553-8

[13] Yang S, Jiang L, Liu Z, Loy C C. Vtoonify: Controllable high-resolution portrait video style transfer. ACM Transactions on Graphics (TOG), 2022, 41(6): 1-15. https://doi.org/10.1145/3550454.3555437

[14] Guo M, Xu F, Wang S, Wang Z, Lu M, Cui X, Ling X. Synthesis, style editing, and animation of 3D cartoon face. Tsinghua Science and Technology, 2023, 29(2): 506-516. https://doi.org/10.26599/TST.2023.9010028

[15] Zhang T, Yu L, Tian S. CAMGAN: Combining attention mechanism generative adversarial networks for cartoon face style transfer. Journal of Intelligent & Fuzzy Systems, 2022, 42(3): 1803-1811. https://doi.org/10.3233/JIFS-211210

[16] Li M. Application of GAN-Based Data Encryption Technology in Computer Communication System. Informatica, 2024, 48(15): 17-34. https://doi.org/10.31449/inf.v48i15.6390

[17] Lobato T, Sottek R, Vorländer M. Using learned priors to regularize the Helmholtz equation least-squares method. The Journal of the Acoustical Society of America, 2024, 155(2): 971-983. https://doi.org/10.1121/10.0024726

[18] Dousoky G M. Classification of Diabetic Retinopathy (DR) using ECA Attention Mechanism

Deep learning Networks. Journal of Advanced Engineering Trends, 2024, 43(1): 425-431. https://doi.org/10.21608/jaet.2022.145091.1210

[19] Yang K, Chang S, Tian Z, Gao C, Du Y, Zhang X, Xue L. Automatic polyp detection and segmentation using shuffle efficient channel attention network. Alexandria Engineering Journal, 2022, 61(1): 917-926. https://doi.org/10.1016/j.aej.2021.04.072

[20] Shin Y, Yoon Y. PGCN: Progressive graph convolutional networks for spatial-temporal traffic forecasting. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(7): 7633-7644. https://doi.org/10.1109/TITS.2024.3349565