Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **10**, Number **2**, May **2025**

Music Notation Recognition Method Based on Improved Generative Adversarial Network

Yang Li*

Huanghe Jiaotong University, Jiaozuo 454000, P. R. China
ly291017791@163.com

Boris Zhang

School of Information Technology and Engineering
St. Paul University Philippines, Tuguegarao 3500, Philippines
bx1609@163.com

*Corresponding author: Yang Li

Received April 9, 2024; revised September 9, 2024; accepted January 9, 2025.

Abstract. *Image fuzzy enhancement is a research hotspot in the field of image processing, which aims to recover enhanced beginning clear images from degraded images. Based on the research of traditional particle swarm optimization algorithm and fuzzy enhancement algorithm, an image fuzzy enhancement method based on membrane computing particle swarm algorithm is proposed. Firstly, in order to make full use of the sparse characteristics of the clear image, the coefficient decomposition under wavelet domain and tightly supported wavelet domain is performed on the image respectively. Then, a joint optimisation model is constructed using L1 parametric constraints to achieve pretzel noise cancellation. Next, the MMH-PSO algorithm is designed by improving the particle swarm algorithm using membrane computing and Metropolis-Hastings sampling. Based on the simulated annealing algorithm temperature drop process, Metropolis-Hastings sampling is used to add randomness to the particle swarm algorithm so that it has the ability to jump out of the local optimum. The use of membrane computing enhances the parallelism of the particle swarm algorithm and can reduce the time complexity in solving complex problems. Finally, MMH-PSO is used to simultaneously search out the magnitude of the two fuzzy parameters in the traditional fuzzy enhancement algorithm in order to improve the accuracy of the algorithm. The experimental results show that the proposed algorithm has better SSIM values than the traditional fuzzy enhancement algorithm, which effectively improves the image quality and makes the image edge information more abundant.*

Keywords: Image enhancement; particle swarm algorithm; membrane computing; simulated annealing algorithm; pretzel noise

1. **Introduction.** The study of music notation recognition has important academic value and practical application significance. From the academic point of view, music notation recognition is essentially a complex image processing and pattern recognition problem [1, 2], which involves image segmentation, feature extraction, symbol recognition and other technical fields. By studying music notation recognition, it can promote the development of related technologies, especially in the fields of deep learning, computer vision and artificial intelligence [3, 4].

From the practical application point of view, music notation recognition technology has a significant contribution to music education, the construction of digital music libraries

and the development of music information retrieval systems. Automatically identifying and converting the notation images into digital format can greatly simplify the digitisation process of music notations and improve the accessibility and usability of music resources [5]. In addition, music notation recognition technology can be applied to intelligent music composition software to assist composers and musicians in their creations, and can even be used for automatic analysis and research of music theory and history [6]. With the continuous progress of technology, music notation recognition is expected to play its value in a wider range of fields and contribute to the innovation and development of the music industry.

Traditional methods for music notation recognition rely on large amounts of high-quality labelled data [7, 8], which are often time-consuming and laborious to acquire. Generative Adversarial Network (GAN) models [9, 10] are capable of generating high-quality music notation images, and these generated images can be used as additional training data to help the models learn more accurate and robust feature representations. Through adversarial training of the generator and discriminator, the generated notation images are not only visually realistic, but also highly consistent with real notationes in terms of structure and content, thus improving the model's ability to maintain the authenticity of musical notation and structure. The aim of this study is to apply GAN in music notation recognition, which not only improves the accuracy of recognition, but also generates high-quality training data, which is also inspiring for understanding and solving image recognition problems in other fields.

1.1. **Related work.** Traditional methods for simple spectrum recognition mainly rely on image processing techniques and pattern recognition algorithms, which are implemented through the steps of image pre-processing, feature extraction, symbol segmentation, symbol recognition and post-processing. These methods have some significant drawbacks when dealing with simple spectrum images, including high requirements on image quality, limitations in feature extraction, recognition accuracy problems, low computational efficiency, and insufficient generalisation capability. In particular, the robustness and accuracy of traditional methods are challenged when confronted with light variations, noise interference, and complex layouts of simple spectra, such as classifiers such as Support Vector Machines (SVMs) and Random Forests.

With the advancement of deep learning techniques, new possibilities are provided to solve these problems, showing higher recognition performance and better generalisation. Niu and Suen [11] proposed a deep Convolutional Neural Network (CNN) model for image recognition task. The model achieved better recognition results on multiple image datasets and achieved significant performance improvement over traditional methods. However, the method has high computational complexity when dealing with large-scale image data and requires a large amount of training data for model training. Bertinetto and Vuorinen [12] introduced the deep residual network (ResNet), a deep convolutional neural network structure with residual learning. This network solves the degeneracy problem in deep networks by allowing gradients to flow directly through the layers, thus achieving higher accuracy. However, ResNet has a large number of layers, resulting in large model parameters that require a large amount of computational resources and data for training. Hu et al. [13] explored the use of Generative Adversarial Networks (GAN) for image style migration. By training a generator to mimic image features of a specific art style, the method is able to convert ordinary photos into images with a specific art style. However, this method has challenges in maintaining content authenticity and stylistic consistency, and sometimes the generated images may be too stylised and lose the accuracy of the original content. Du et al. [14] proposed an image recognition method based on the attention

mechanism, which enhances the model's ability to recognise key features by assigning different attention weights to different regions of the image. Although the method improves the accuracy of recognition, the computational complexity of the attention model is high and may not be suitable for deployment in resource-constrained environments. Zhang et al. [15] proposed a deep learning framework that combines local and global features for improving the performance of image classification. The framework enhances the understanding of image content by processing local region and global image information in parallel. However, the method has high memory and storage requirements when processing large-scale image datasets, which may limit its application in resource-constrained environments. Wen et al. [16] explored an image recognition method based on self-supervised learning, which does not rely on labelled data but uses the structural information of the image itself to guide model learning. Through self-supervised learning, the model is able to learn useful feature representations without explicit labelling. Although this method reduces the dependence on labelled data, its recognition effect in complex scenes still falls short of that of supervised learning methods and requires high quality of pre-training data.

1.2. **Motivation and contribution.** Musical scores have a variety of layouts and styles, including different notes, rests, ornaments, chords and other elements, as well as their different arrangements and combinations on the pentatonic score. Therefore, the traditional GAN model has the problem of poor adaptability when dealing with music notationes of different styles and sources. In order to solve the above problems, an improved GAN model is proposed, which can better focus on and learn the key symbols and their spatial relationships in a simple score, thus improving the adaptability and recognition accuracy for complex simple score layouts. The main innovations and contributions of this work include: (1) In the generative network, residual concatenation is introduced to enhance feature learning and image detail capture. And in the Discriminator network, average pooling layer and deep asymmetric convolution are used to enhance the Discriminator accuracy. (2) In order to enhance the model's understanding of the relationships between different symbols in a music notation image, a multi-head attention mechanism module is introduced into the generator and discriminator to capture richer contextual information. In addition, in order to enable the model to capture the spatial location information of the symbols in the music notation image, location coding is introduced in the attention module. (3) In the improved GAN model, the generator and discriminator are trained by combining the adversarial loss, feature matching loss and content loss. In addition, the Adam optimiser, which combines momentum estimation and adaptive learning rate adjustment, is used for parameter updating to effectively deal with complex non-convex optimisation problems, thus ensuring the stability and efficiency of model training.

2. **Relevant concepts and principles.**

2.1. **Overview of music notation recognition technology.** Music notation recognition technology aims to convert music notations in image format into editable and searchable digital format, thus facilitating the management and processing of music information. The technology usually involves several fields such as image processing, pattern recognition and machine learning. The core tasks of the short score recognition technique include steps such as image preprocessing, feature extraction, symbol segmentation, recognition and post-processing.

(1) Image Preprocessing and Feature Extraction

Prior to the recognition process, it is first necessary to pre-process the parsimony image,

including steps such as denoising, binarisation, and skew correction, in order to facilitate subsequent feature extraction. For example, denoising using Gaussian filter can be expressed as

$$I'(x,y) = G * I(x,y) \tag{1}$$

where $I'(x,y)$ is the denoised image, $I(x,y)$ is the original image, and G is the Gaussian kernel function.

Feature extraction aims at extracting information from preprocessed images that contributes to symbol recognition. This usually involves edge detection, contour extraction, morphological operations, etc. For example, edge detection using the Sobel operator can be expressed as

$$\nabla I(x,y) = \begin{bmatrix} \frac{\partial I}{\partial x}(x,y) \\ \frac{\partial I}{\partial y}(x,y) \end{bmatrix} \approx \begin{bmatrix} 0 & -1 & 0 \\ -2 & 4 & -2 \\ 0 & -1 & 0 \end{bmatrix} * I(x,y) \tag{2}$$

where $\nabla I(x,y)$ is the gradient of the image and $*$ denotes the convolution operation.

(2) Symbol segmentation and recognition

Each symbol (e.g., note, rest, bar line, etc.) in a simple score needs to be accurately segmented and recognised. Symbol segmentation can be done using a projection method, where the horizontal and vertical projection histograms of the image are computed to locate the symbol positions. The horizontal projection P_h and the vertical projection P_v can be expressed as follows, respectively.

$$P_h(i) = \sum_{j=1}^{H} I(i,j) \tag{3}$$

$$P_v(j) = \sum_{i=1}^{W} I(i,j) \tag{4}$$

where $I(i,j)$ is an element in the image matrix; W and H are the width and height of the image, respectively; $P_h(i)$ and $P_v(j)$ denote the pixel sums of the i-th row and the j-th column, respectively.

The recognition process involves matching the segmented symbols with known notation symbols to determine the musical meaning they represent. This can be achieved by template matching, pattern recognition or deep learning methods. For example, the use of a cross-correlation matching template can be represented as:

$$S_{\text{match}} = \max_{\tau} \sum_{x,y} \big[T(x,y) * R(x-\tau, y)\big] \tag{5}$$

where $T(x,y)$ is the template image, $R(x,y)$ is the image of the symbol to be recognised, and τ is the possible translation offset.

(3) Post-recognition processing and correction

The recognised symbols need to be corrected and restructured according to music theory to ensure that the converted short notes are accurate. For example, note sequences can be modelled and corrected using a Hidden Markov Model (HMM). The state transfer probabilities of the HMM are expressed as follows.

$$a_{ij} = P\big(q_{t+1} = s_j \mid q_t = s_i\big) \tag{6}$$

where a_{ij} is the probability of transferring from state s_i to state s_j and q_t is the hidden state at moment t.

2.2. **Principles of Generative Adversarial Networks.** A Generative Adversarial Network (GAN) is a deep learning model consisting of two competing neural networks, a Generator and a Discriminator [17, 18]. The schematic diagram of a GAN usually contains two neural network structures, one acting as a Generator and the other as a Discriminator. The generator receives a random noise vector and generates data, while the discriminator tries to distinguish the generated data from the real data. These two networks compete with each other during the training process, constantly adjusting their parameters to improve performance, as shown in Figure 1 [19].

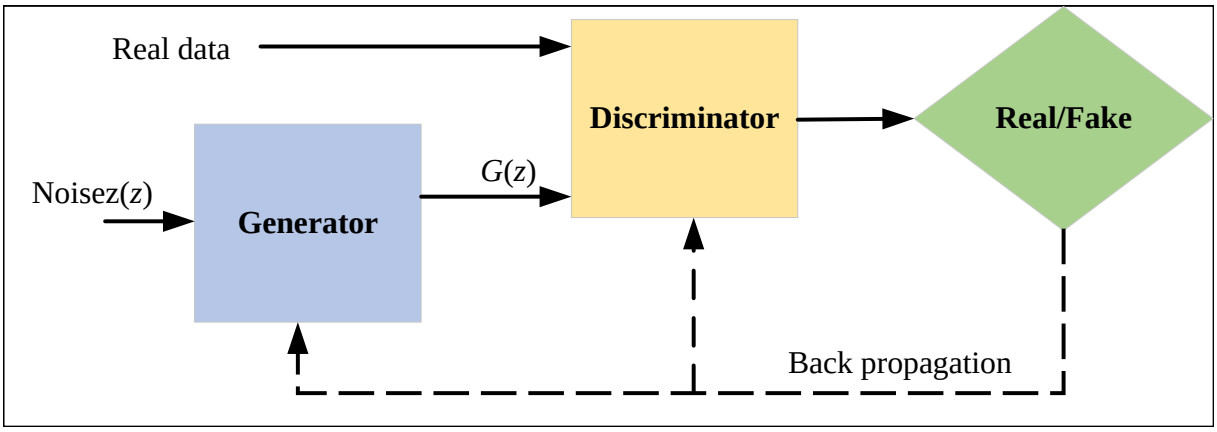

Figure 1. Typical GAN structure

The core idea of GAN is to improve the quality of the generated data by generating the data through a generator and evaluating the data by a discriminator, both of which work against each other. The objective of the generator is to generate fake data that is as close as possible to the real data distribution. The objective function can be expressed as [20]:

$$\min_G \mathbb{E}_{x\sim p_x(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{7}$$

where $p(x)$ is the distribution of the real data, $p_z(z)$ is the distribution of the random noise, $D(x)$ is the discriminator's evaluation of the real data, and $G(z)$ is the data generated by the generator based on the noise.

The objective of the discriminator is to distinguish between real data and fake data produced by the generator [21]. Its objective function can be expressed as:

$$\min_D \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{8}$$

This is the same as the objective function of the generator, since the GAN is trained by alternately optimising the generator and the discriminator.

The training of a GAN is viewed as a maximum mean–minimum–maximum (minimax) game problem [22], where the goal is to find an equilibrium that optimises the performance of the generator and discriminator.

$$\min_{G\in\mathcal{G}} \max_{D\in\mathcal{D}} V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{9}$$

Alternating training between Gradient Ascent and Gradient Descent is required [23, 24]. During the training of GAN, the parameters of the generator and discriminator are optimised by alternating between Gradient Ascent and Gradient Descent.

$$G_{t+1} = G_t + \alpha\nabla_G \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{10}$$

$$D_{t+1} = D_t - \beta\nabla_D \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{11}$$

where α and β are the learning rates, and G_t and D_t are the generator and discriminator after the t-th iteration, respectively.

$$\min_G \mathbb{E}_{x\sim p(x),\, z\sim p_z(z)}\left[\|\phi(x) - \phi(G(z))\|^2\right] \tag{12}$$

where $\phi(x)$ is the feature representation function extracted from the discriminant.

3. **Improved Generative Adversarial Network Models.**

3.1. **Model architecture.**

Our proposed improved generative adversarial network model aims to improve the quality of recognition and generation of music notation images. The model creates realistic notation images by means of a well-designed generative network, while the authenticity of the generated images is evaluated using a Discriminator network. In the generative network, we introduce Residual Connectivity (RC) and attention mechanisms to enhance feature learning and image detail capture. While in the Discriminator network, we employ average pooling layer and Deep Asymmetric Convolution (DSC) to enhance the Discriminator accuracy. In addition, by optimising the loss function and adopting an effective optimisation strategy, our model achieves high performance in the music notation recognition task, generating high-quality images that are virtually indistinguishable from real scores while accurately distinguishing between real and generated score images. This improved model not only performs well in the digitisation of music notations, but also provides new solutions for image generation and recognition tasks in related fields.

3.1.1. *Improved RC-based Generator Network.* The Generator plays the role of a creator in the improved GAN, whose goal is to generate fake images that are indistinguishable from real music notation images. In order to better capture the features of music notations, the design of a generative network usually includes the following key components:

(1) Initial layer: receives a random noise vector as input and maps it to a fixed-size feature vector [25] via a fully connected layer as follows:

$$h_0 = \sigma\big(W_0 z + b_0\big) \tag{13}$$

where h_0 is the initial feature map; W_0 and b_0 are the weights and biases of the fully connected layer; and σ is the ReLU activation function.

(2) Downsampling layer: the spatial dimension of the feature map is gradually reduced by convolutional and pooling layers, while the number of feature channels is increased to capture more abstract image features, as follows:

$$h_{\text{pool}} = \text{pool}\big(f(h_{\text{prev}}, W_{\text{conv}}, b_{\text{conv}})\big) \tag{14}$$

where h_{prev} is the feature map of the previous layer; W_{conv} and b_{conv} are the weights and bias of the convolution layer; f denotes the convolution operation; pool denotes the pooling operation.

(3) Residual connectivity: introducing RC in the deeper layers of the network to facilitate the back propagation of the gradient, alleviate the gradient vanishing problem, and improve the training efficiency of the network.

$$h_{\text{res}} = h_{\text{skip}} + \text{ReLU}\big(h_{\text{prev}} \cdot W_{\text{res}} + b_{\text{res}}\big) \tag{15}$$

where h_{res} is the output of the residual block; h_{skip} is the direct connection that skips one or more layers; W_{res} and b_{res} are the weights and biases in the residual block.

(4) Upsampling layer: the spatial dimensions of the feature map are gradually recovered by transposing the convolutional layer, while the number of feature channels is reduced to reconstruct the details of the image.

$$h_{\text{up}} = \text{upConv}\big(h_{\text{prev}}, W_{\text{up}}, b_{\text{up}}, \text{stride}, \text{padding}\big) \tag{16}$$

where h_{up} is the feature map after upsampling; upConv denotes the transpose convolution operation; stride and padding are the step and padding parameters, respectively.

(5) Output layer: the last convolutional layer generates the final notation image using an activation function (Tanh) in pixel space.

$$G(z) = \tanh\big(W_{\text{out}}\, h_{\text{last}} + b_{\text{out}}\big) \tag{17}$$

where $G(z)$ is the generated simple spectrum image; h_{last} is the feature map of the last layer; W_{out} and b_{out} are the weights and biases of the output layer; tanh is the activation function of the output layer. Limiting the output values to the range of $[-1, 1]$ simulates the pixel intensity of the simplex image.

3.1.2. *Improved Discriminator network based on DSC.* The main purpose of the Discriminator network is to distinguish whether the input image is derived from real data or data output from the generative network. In this paper, on the basis of the original Discriminator network structure, DSC is used instead of the traditional convolution, and then the features of the input image are extracted by six convolution modules, highlighting the important information in the feature map, which improves the performance of image reconstruction. The improved discriminant network structure is shown in Figure 2.

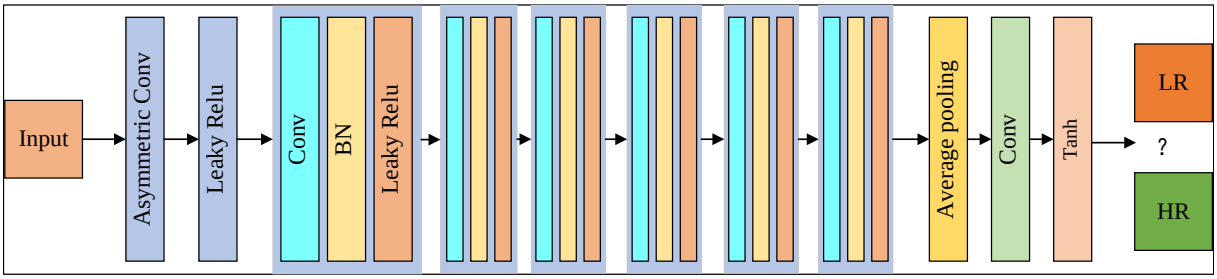

Figure 2. Typical GAN structure

In order to make the training of the network more stable and reduce the amount of network parameter computation, parallel 3×3, 3×1, and 1×3 asymmetric convolutions are chosen instead of the traditional convolution, as shown in Figure 3. Then, the features of the input image are extracted by six convolutional modules consisting of convolutional layers, batch normalisation layers, and activation functions. In order to avoid overfitting phenomenon in network training, average pooling layer is used instead of fully connected layer to calculate the average value of feature image element in each layer. The tanh activation function is then used to output the classification results of the network on the samples, and the input data source is discriminated based on the results. The improved Discriminator network makes the weight parameter change more sensitive and improves the network degradation problem.

3.2. **MultiHeadAttention (MHA).** In GAN, the attention mechanism is crucial for improving the model's ability to capture key features of music notation images. By introducing an attention module, the model is able to mimic the focusing properties of human vision and concentrate on learning the regions that are most important for the recognition task.

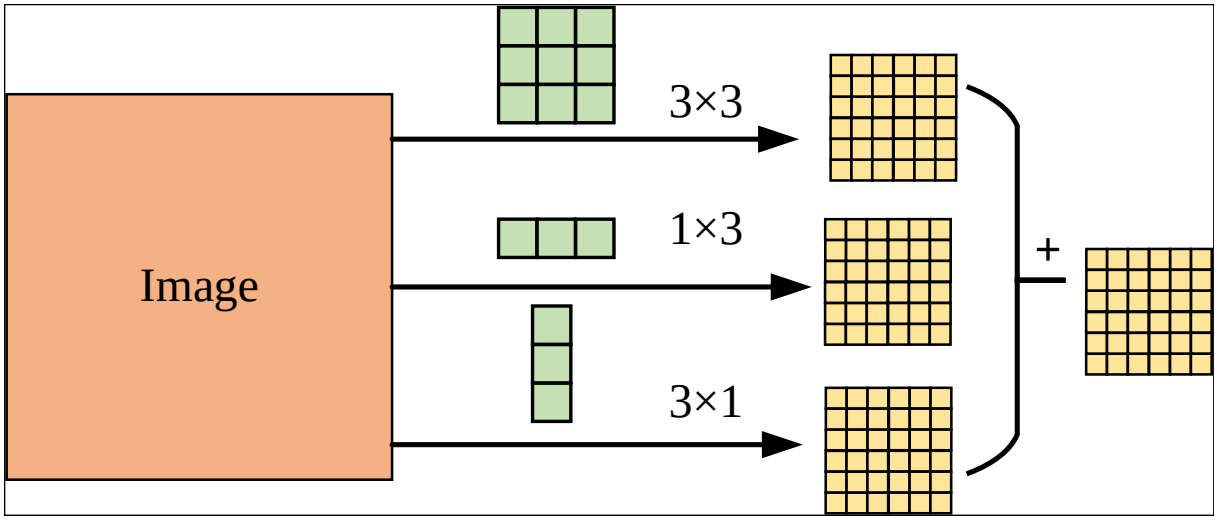

Figure 3. Asymmetric convolutional structures

In order to enhance the model's understanding of the relationships between different symbols in a music notation image, we introduced the MHA module into the generator and discriminator [26, 27]. This module enables the model to capture richer contextual information by learning multiple attentional weights in parallel, allowing the model to focus on different locations and scales of the image at the same time. The computation of MHA can be expressed as:

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\, W^O \tag{18}$$

where Q, K, and V are the query, key, and value of the input features, respectively; W^O is the output weight matrix; h is the number of attention headers; and Concat denotes concatenating the outputs of multiple headers.

To enable the model to capture the spatial location information of the symbols in the music notation image, we introduce positional coding in the attention module. The method for generating the position encoding is shown below:

$$\text{PosEncoding}(P) = P \cdot \exp\left(2\pi i \frac{p}{P_{\max}}\right) \tag{19}$$

where P is the position encoding matrix, p is the position index, $P_{\max}$ is the maximum position index, and i is the imaginary unit.

With the above approach, our improved generative adversarial network model can effectively incorporate the attention mechanism, which not only improves the generation quality of music notation images, but also enhances the model's ability to recognise key features in the notation images.

3.3. **Integrated loss function and optimisation strategy.** In the improved GAN model, designing an appropriate loss function and adopting an effective optimisation strategy are crucial to enhance the model performance. In the improved GAN model, we train the generator and discriminator by integrating the adversarial loss, feature matching loss and content loss.

The confrontation loss drives the generator to produce realistic notation images, while prompting the discriminator to accurately distinguish between true and false images. The feature matching loss ensures that the generated image matches the real image at the feature level, while the content loss ensures the consistency of the high-level feature structure. The total loss function optimises the model performance by balancing these

loss terms in an integrated manner. In addition, parameter updates are performed using the Adam optimiser, which combines momentum estimation and adaptive learning rate tuning to effectively deal with complex non-convex optimisation problems, thus ensuring stability and efficiency in model training. Together, these strategies make the generated notation images highly similar to real notation images both visually and structurally, which greatly improves the performance of the music notation recognition and generation task.

Adversarial loss is central to GAN training and is used to measure the adversarial nature between the generator and the discriminator. We use a variant of the Wasserstein GAN [28, 29] with an adversarial loss function defined as:

$$L_{\text{adv}}(G, D) = -\mathbb{E}_{x\sim p}[D(x)] + \mathbb{E}_{z\sim p_z}[D(G(z))] \tag{20}$$

This loss function encourages the generator to produce images that can deceive the discriminator, while the discriminator tries to correctly distinguish the real image from the generated image.

In order to make the generated notation image closer to the real image at the feature level, we introduce the feature matching loss, which is computed by comparing the outputs of the generated image and the real image at the discriminator feature extraction layer.

$$L_{\text{feat}}(G, D) = \mathbb{E}_{x\sim p}\big[\|\phi(x) - \phi(G(z))\|^2\big] \tag{21}$$

where ϕ is the feature representation function extracted from the discriminator and z is the random noise vector.

Content loss is used to ensure that the generated notation image is consistent with the real image in terms of content. Typically, we compute the content loss by comparing the feature activations of both at some intermediate layer.

$$L_{\text{content}}(G) = \mathbb{E}_{z\sim p_z,\, x\sim p}\big[\|\phi(x) - \phi(G(z))\|^2\big] \tag{22}$$

This helps the generator to learn the high-level feature structure of the real notation image. The total loss function is a combination of the losses of the above components, which combines the requirements of adversarial, feature matching and content consistency.

$$L_{\text{total}}(G, D) = L_{\text{adv}}(G, D) + \lambda_{\text{feat}}\, L_{\text{feat}}(G, D) + \lambda_{\text{content}}\, L_{\text{content}}(G) \tag{23}$$

where λ_{feat} and λ_{content} are the weight coefficients used to balance the different loss terms.

In order to efficiently optimise the above loss function, we use the Adam optimiser, which combines the features of momentum estimation and adaptive learning rate for complex non-convex optimisation problems. The parameters of the generator and the discriminator are updated in the following way:

$$\theta_{G,t+1} = \theta_G + \alpha\, \nabla_{\theta_G} L_{\text{total}}(G, D) \tag{24}$$

$$\theta_{D,t+1} = \theta_D + \alpha\, \nabla_{\theta_D} L_{\text{total}}(G, D) \tag{25}$$

where θ_G and θ_D are the parameters of the generator and discriminator, respectively; and α is the learning rate.

With these well-designed loss functions and optimisation strategies, our improved GAN model is able to efficiently learn to generate high-quality music notation images while ensuring that these images remain highly visually and structurally consistent with real notation images.

4. **Music notation datasets and score recognition.**

4.1. **Dataset description.** In order to train and evaluate the performance of the improved generative adversarial network model in the music notation recognition task, we constructed a dataset containing rich and diverse music notation images. The resolution of the images in the dataset is 1024 × 768 pixels and there are 200 songs. The dataset contains music notation images of various styles and complexity, ranging from simple monophonic melodies to complex polyphonic compositions, ensuring that the model is able to learn a wide range of features of music notations. Examples include classical, pop, and ethnic music, with each style containing at least 1,00 images. The total number of score images contained in the dataset is 10,00 images. Each image is equipped with detailed annotation information, including the location and attributes of musical elements such as notes, rests, bar lines, key signatures, beat signatures, etc., which facilitates the model's learning and understanding of the structure and meaning of musical notation. The dataset has a sufficient sample size to cover the diversity of musical notationes and reduce the risk of overfitting, while providing sufficient data to support the generalisation capability of the model.

4.2. **preprocessing process.** The preprocessing process is an important step in music notation recognition, which lays a solid foundation for subsequent feature extraction and model training. Our preprocessing process includes the following key steps:

(1) Grayscaling: converting a colour notation image into a grayscale image to reduce computational complexity while retaining enough information for symbol recognition.

(2) Binarisation: clear separation of the simple symbols from the background by converting the grey scale image to a binary image using the weighted average method.

(3) Denoising: apply Gaussian filter to remove noise and abnormal pixels in the image to improve the image quality.

(4) Tilt Correction: detect and correct the tilt in the image based on the Hough Transform method to ensure that all the short-spectrum images are in a horizontal state, which is convenient for subsequent processing.

(5) Symbol segmentation: symbol segmentation can be done using projection method [30, 31], where the symbol positions are localised by calculating the horizontal and vertical projection histograms of the image.

4.3. **Data enhancement and parsimony identification.** In order to improve the generalisation ability and robustness of the model, a series of data enhancement methods were used to expand the dataset and simulate more variations. The geometric transformation parameters include a range of rotation angles (–10 to 10 degrees), a range of scaling ratios (90 % to 110 %), and a range of translation distances (1 % to 5 % of the image width). By changing the brightness and contrast of the image, images under different lighting conditions are simulated to enhance the model's adaptability to lighting changes. The range of brightness adjustment is –50 to +50.

Ultimately, the steps of the music notation recognition method based on the proposed improved GAN model are as follows:

Step 1: Collect a dataset containing images of musical notations of various styles and complexity and perform pre-processing operations such as grayscaling, binarisation, denoising and skew correction on the images to improve the accuracy of subsequent processing;

Step 2: Construct the Generator network, including the initial layer, downsampling layer, RC, upsampling layer, and output layer, to generate high-quality simple spectral images;

Step 3: Construct Discriminator including input layer, convolutional layer, pooling layer and output layer to differentiate between real notation image and generated notation image;

Step 4: Introduce MHA modules in the generative and Discriminator networks to enhance the ability of the model to focus on key features;

Step 5: Optimisation of model parameters by back propagation algorithm through adversarial loss, feature matching loss and content loss in order to optimise the performance of the generator and discriminator in a combined manner and using Adam optimiser and appropriate learning rate tuning strategy.

Step 6: Optimise the model parameters by back-propagation algorithm using Adam optimiser and appropriate learning rate tuning strategy.

Step 7: Identify and reconstruct the new notation image using the trained Generative Adversarial Network model. The authenticity of the generated notation image is evaluated by the Discriminator network and the musical notation and structural information is extracted.

5. **Experimental design and evaluation.**

5.1. **Experimental setup.** The experimental environment is based on the Pytorch deep learning framework under Windows 10 with a graphics card model NVIDIA GeForce RTX 3080. The hardware parameters of the model running platform are shown in Table 1. The experimental dataset is as described in Section 4.1. The Adam optimiser is used, the learning rate is initialised to 0.0003, and the learning rate is reduced to 1/10 of the original one halfway through the training. The Epoch of the model is 50 times and the batch size is 64.

Table 1. Hardware parameters of the model runtime platform

Parameters	Descriptions
CPU Model	Intel Core i7-10700K
CPU core	8 cores (16 threads)
CPU frequency	3.8 GHz (maximum capacity 4.8 GHz)
Memory capacity	32 GB DDR4
storage capacity	1 TB SSD + 2 TB HDD
Graphics card model	NVIDIA GeForce RTX 3080
memory capacity	10 GB GDDR6
operating system	Windows 10

As an example, some of the pentatonic and simple scores of "The Little Snail" are shown in Figure 4. Preprocessing is performed before the recognition of the short score using the improved GAN model, and symbol segmentation is performed using the projection method to locate the symbol positions.

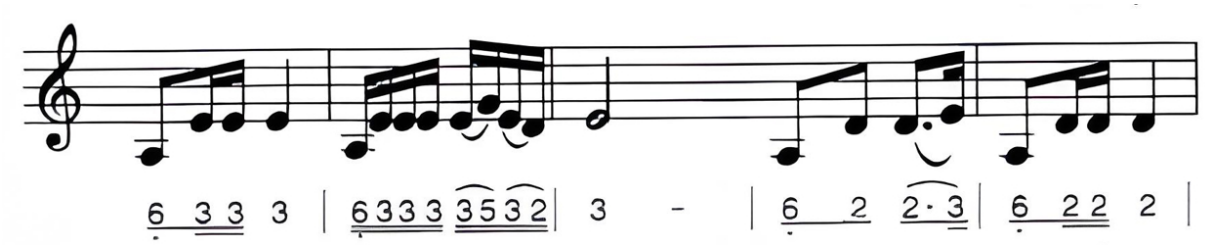

Figure 4. Partial five-line score and short score of "The Little Screwtape".

5.2. **Ablation experiment.** Ablation experiments are conducted to verify the effectiveness of RC, MHA, Comprehensive Loss Function (CLF) and Optimisation Strategy (OS) in the proposed improved GAN model for music notation recognition. GAN is a traditional basic network architecture model; RC model is to add residual connections to the generative network of GAN model; MHA model is to add multi-head attention module on the basis of RC model; CLF model is to introduce comprehensive loss function on the basis of MHA model; OS model is to add optimisation strategy on the basis of MHA model. All models are trained 50 times. Comparisons are made from the five dimensions of the models—Loss, Accuracy, Precision, Recall and F1-score—and the results are shown in Figure 5 and Figure 6.

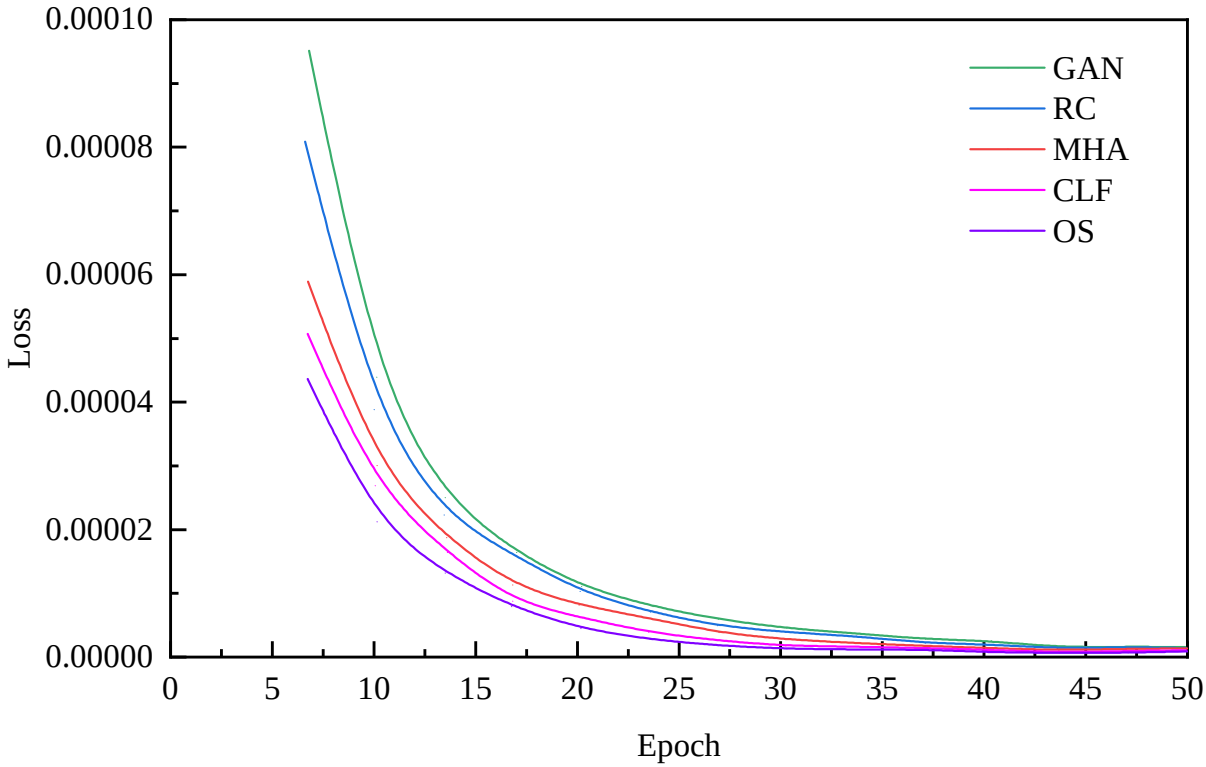

Figure 5. Modelling losses

It can be seen that the GAN model has a medium convergence speed, and each evaluation index stays around 99.960%, but the number of parameters is higher. The number of parameters of the RC model is significantly reduced to 16.6% of the CNN model, and the evaluation indexes such as precision rate and recall rate are slightly lost, but the evaluation indexes can still be maintained at more than 99.955 %. The MHA model has a faster convergence speed than that of the CNN model, and the other evaluation indexes are all better than those of the above models, which indicates that the use of the MHA can greatly improve the model recognition performance. The CLF model significantly improves the evaluation indexes to more than 99.970%, and the OS model has the fastest convergence speed and the best accuracy, recall, precision and F1-score indexes, which are more than 99.975%.

5.3. **Model comparison.** In order to verify the superiority of the improved GAN model proposed in this paper, it is compared with BPNN, CNN, RNN and GAN on the music notation dataset, and the results are shown in Table 2.

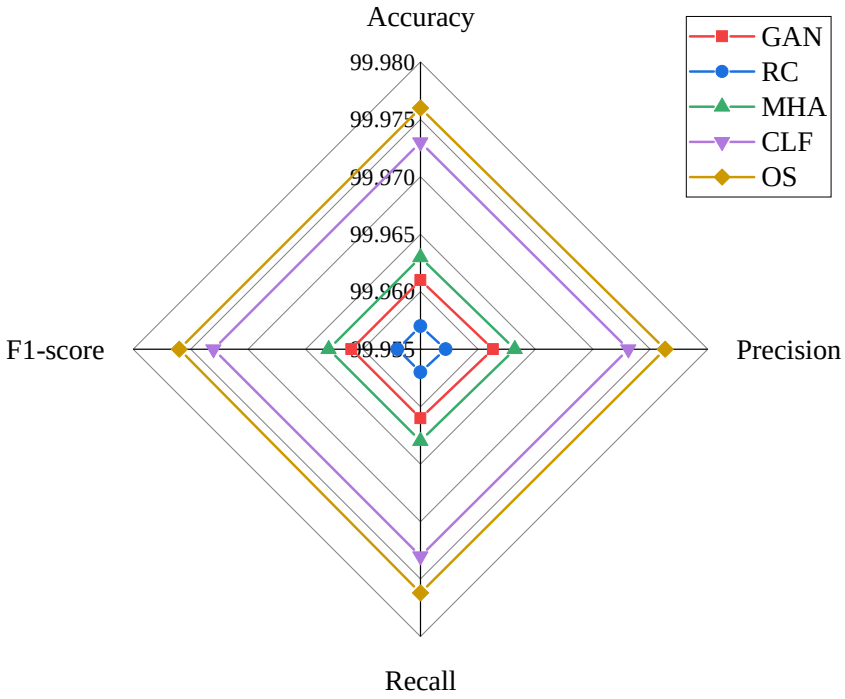

Figure 6. Evaluation indicators

Table 2. Comparative results of models

Modelling	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
BPNN	99.72	99.72	99.72	99.72
CNN	99.77	99.77	99.77	99.77
RNN	99.83	99.83	99.83	99.83
GAN	99.92	99.92	99.94	99.91
Ours	99.98	99.98	99.98	99.98

The conventional BPNN achieves 99.72% in all the metrics, which is a fairly good result, indicating that the BPNN is able to handle the music notation recognition task effectively. The performance of CNN is slightly improved with 99.77% in all the metrics, which may be due to the fact that CNN can better capture local features and is suitable for image-related tasks. The performance of RNN is further improved with 99.83% in all the metrics. RNN is especially suitable for dealing with sequential data, and music notation is essentially a kind of sequential information, so RNN has some advantages in dealing with this kind of data. The standard GAN achieves a performance of 99.92%, exceeding the previous models in accuracy, precision, recall and F1-score. The GAN is able to learn more complex data distributions through adversarial training and generate more realistic images, which may help to improve the recognition performance. Finally, the improved GAN model proposed in this paper achieves 99.98% on all metrics, which is the best result among all models. This result shows that the improved GAN model can effectively improve the accuracy and robustness of music notation recognition by incorporating the attention mechanism, optimal loss function and optimisation strategy.

In summary, the improved GAN model demonstrates superior performance on the music notation recognition task, which is related to its ability to better capture and process key features of music notation images.

6. **Conclusions.** In this work, an improved GAN model is proposed so as to increase the adaptability and recognition accuracy for complex notation layouts. In the generative network, residual concatenation is introduced to enhance feature learning and image detail capture. While in the Discriminator network, average pooling layer and deep asymmetric convolution are used to enhance the Discriminator accuracy. A multi-head attention mechanism module is introduced in the generator and discriminator to capture richer contextual information, and positional encoding is introduced in the attention module. In the improved GAN model, the generator and discriminator are trained by integrating the loss function. In addition, the Adam optimiser, which combines momentum estimation and adaptive learning rate tuning, is used for parameter updating, thus ensuring the stability and efficiency of model training. Experimental results show that the proposed improved GAN model achieves 99.98% in all metrics. Future research can further explore how to improve the generalisation ability of the proposed models and how to apply these models to practical music education and music information processing systems.

REFERENCES

[1] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, "Understanding optical music recognition," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.

[2] M. Alfaro-Contreras and J. J. Valero-Mas, "Exploiting the two-dimensional nature of agnostic music notation for neural optical music recognition," *Applied Sciences*, vol. 11, no. 8, p. 3621, 2021.

[3] D. Bainbridge and T. Bell, "A music notation construction engine for optical music recognition," *Software: Practice and Experience*, vol. 33, no. 2, pp. 173–200, 2003.

[4] M. Alfaro-Contreras, A. Ríos-Vila, J. J. Valero-Mas, J. M. Iñesta, and J. Calvo-Zaragoza, "Decoupling music notation to improve end-to-end Optical Music Recognition," *Pattern Recognition Letters*, vol. 158, pp. 157–163, 2022.

[5] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Handwritten music recognition for mensural notation with convolutional recurrent neural networks," *Pattern Recognition Letters*, vol. 128, pp. 115–121, 2019.

[6] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, pp. 95–121, 2001.

[7] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.

[8] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, p. 40, 2019.

[9] F. Zhang, T.-Y. Wu, and G. Zheng, "Video salient region detection model based on wavelet transform and feature comparison," *EURASIP Journal on Image and Video Processing*, vol. 2019, p. 58, 2019.

[10] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[11] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.

[12] C. G. Bertinetto and T. Vuorinen, "Automatic baseline recognition for the correction of large sets of spectra using continuous wavelet transform and iterative fitting," *Applied Spectroscopy*, vol. 68, no. 2, pp. 155–164, 2014.

[13] H. Hu, S. Li, Z. Qian, and X. Zhang, "Domain transferred image recognition via generative adversarial network," *Security and Communication Networks*, vol. 2022, p. 2015, 2022.

[14] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1347–1360, 2017.

[15] J. Zhang, Y. Xie, Q. Wu, and Y. Xia, "Medical image classification using synergic deep learning," *Medical Image Analysis*, vol. 54, pp. 10–19, 2019.

[16] Z. Wen, Z. Liu, S. Zhang, and Q. Pan, "Rotation awareness based self-supervised learning for SAR target recognition with limited training samples," *IEEE Transactions on Image Processing*, vol. 30, pp. 7266–7279, 2021.

[17] H. Emami, M. M. Aliabadi, M. Dong, and R. B. Chinnam, "Spa-gan: Spatial attention gan for image-to-image translation," *IEEE Transactions on Multimedia*, vol. 23, pp. 391–401, 2020.
[18] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "Gan inversion: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3121–3138, 2022.
[19] A. You, J. K. Kim, I. H. Ryu, and T. K. Yoo, "Application of generative adversarial networks (GAN) for ophthalmology image domains: a survey," *Eye and Vision*, vol. 9, no. 1, p. 6, 2022.
[20] Z. Qin, Z. Liu, P. Zhu, and Y. Xue, "A GAN-based image synthesis method for skin lesion classification," *Computer Methods and Programs in Biomedicine*, vol. 195, p. 105568, 2020.
[21] S. Azadi, D. Pathak, S. Ebrahimi, and T. Darrell, "Compositional gan: Learning image-conditional binary composition," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2570–2585, 2020.
[22] S. Niu, B. Li, X. Wang, and H. Lin, "Defect image sample generation with GAN for improving defect recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1611–1622, 2020.
[23] R. Nandhini Abirami, P. Durai Raj Vincent, K. Srinivasan, U. Tariq, and C.-Y. Chang, "Deep CNN and deep GAN in computational visual perception-driven image analysis," *Complexity*, vol. 2021, pp. 1–30, 2021.
[24] J. Wang et al., "CA-GAN: Class-condition attention GAN for underwater image enhancement," *IEEE Access*, vol. 8, pp. 130719–130728, 2020.
[25] J. Ma, W. Yu, C. Chen, P. Liang, X. Guo, and J. Jiang, "Pan-GAN: An unsupervised pan-sharpening method for remote sensing image fusion," *Information Fusion*, vol. 62, pp. 110–120, 2020.
[26] J. Li, X. Wang, Z. Tu, and M. R. Lyu, "On the diversity of multi-head attention," *Neurocomputing*, vol. 454, pp. 14–24, 2021.
[27] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, and L. B. M. Neti, "Sarcasm detection using multi-head attention based bidirectional LSTM," *IEEE Access*, vol. 8, pp. 6388–6397, 2020.
[28] J. Engelmann and S. Lessmann, "Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning," *Expert Systems with Applications*, vol. 174, p. 114582, 2021.
[29] Y. Zhang, Y. Li, Y. Zhu, and X. Hu, "Wasserstein GAN based on Autoencoder with back-translation for cross-lingual embedding mappings," *Pattern Recognition Letters*, vol. 129, pp. 311–316, 2020.
[30] X.-L. Yun, Y.-M. Zhang, F. Yin, and C.-L. Liu, "Instance GNN: a learning framework for joint symbol segmentation and recognition in online handwritten diagrams," *IEEE Transactions on Multimedia*, vol. 24, pp. 2580–2594, 2021.
[31] K. Kaiyrbekov and M. Sezgin, "Deep stroke-based sketched symbol reconstruction and segmentation," *IEEE Computer Graphics and Applications*, vol. 40, no. 1, pp. 112–126, 2019.