Far Western Review

A Multidisciplinary, Peer Reviewed Journal
ISSN: 3021-9019
Published by Far Western University
Mahendranagar, Nepal

Generative Adversarial Network Based Music Generation

Abhishek Deupa*, Garima Saha, Nikhil Sharma and Virendra Pal Singh
Department of Computer Science and Engineering
Sharda University
Greater Noida, India
*Corresponding author email: abhishekdeupa@gmail.com

Abstract

Music has been an integral part of human civilization personally and culturally. Historically, music has been generated using various instruments, or natural sounds like water drops, or unconventional musical instruments like metal or glass-wares. At present, technologies like Musical Instrument Digital Interface (MIDI) are used to generate music electronically. This research investigates the use of Generative Adversarial Networks (GANs) for beginner-friendly music production. This model uses Long Short Term Memory (LSTM) generator and Patch GAN as discriminator for the GAN architecture. The generator consists of input layer, embedding layer, LSTM layer and generates output with a softmax function. Similarly, the discriminator consists of a convolution layer, the output of which is averaged by the global average pooling layer, and output is generated by the sigmoid function. The model is trained on Maestro MIDI dataset.We make the process understandable by delving into the implementation specifics and outlining the fundamental concepts of music. Our effective model highlights the potential of GANs in music composition by producing cohesive music. After training for 50 epochs, the model exhibited a remarkable precision of 91.82 percent. This project uses the combination of Artificial Intelligence (AI) with music theory to provide intriguing new opportunities in the field of music. The model can be beneficial for different industries like gaming, music, entertainment, education, etc.

Keywords: Generative adversarial networks, music generation, artificial intelligence, long short term memory, patch discriminator

Introduction

Using generative models and human cues, generative artificial intelligence (AI) is a state-of-the-art technology that can generate text, pictures, music, and other types of media material. Generative AI had a tremendous increase in popularity between 2022 and 2023, with a wide range of uses ranging from chatbots to AI-powered motion pictures (AlDahoul et al., 2023). Generative AI models like Codex, DALL-E, and ChatGPT have been receiving a lot of attention from the public. Artificial Intelligence Generated Content (AIGC) includes the use of Generative AI (GAI) approaches to create digital content using AI models, including music, graphics, and natural language. The objective of AIGC is to expedite the generation of high-quality content by streamlining and simplifying the process of content creation. AIGC is accomplished by taking human instructions, deriving meaning from them, and using that intended information to generate content based on its knowledge and understanding (Cao et al., 2023). These technologies' generative powers will probably drastically change the ways in which artists generate ideas and bring them to life. Instead of portending the end of art, generative AI is a new medium with unique affordances of its own (Epstein & Hertzmann, 2023).

Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Latent Dirichlet Allocation (LDA), Restricted Boltzmann Machines (RBM), Deep Belief Networks (DBN), and Deep Boltzmann Machines (DBM) are only a few of the several types of Generative AI models. One of them is the Generative Adversarial Network (GAN), whose development has made it feasible to employ AI to produce highly realistic works of music, art, and other media (GM et al., 2020). GANs offer a method for learning deep representations without a lot of labeled training data. They do this by using a competitive approach utilizing two networks to derive backpropagation signals. Many applications, such as image synthesis, semantic image editing, style transfer, picture super-resolution, and classification, can make use of the representations that GANs can learn (Goodfellow et al., 2020).

GANs were also created to address the generative modeling issues like statistical error and failure to converge to exactly the optimal parameters. A generative model looks at a set of training samples and aims to identify the probability distribution that produced each one. The calculated probability distribution may then be used to create new instances using Generative Adversarial Networks (GANs). Computational benefits of GANs include the elimination of the necessity for Markov chains, the elimination of inference during learning, and the capacity to include a large range of functions in the model. The fact that input elements are not directly replicated into the generator's parameters is another benefit of using a GAN, which may provide statistical benefit to them. The ability to depict extremely crisp, even degenerate distributions is another benefit of adversarial networks over Markov chain-based approaches, which require a slightly blurry distribution in order for the chains to be able to mix between modes (Alqahtani et al., 2019).

The use of GANs for the creation of images has grown within the last several years. Music creation is a good application for GANs because of the unpredictable and random nature of music structure (Creswell et al., 2017).

The combination of creativity and innovation in the field of music and technology has the potential to completely change how producers, composers, and music lovers perceive music. The paper is driven by a great love of artificial intelligence and music, as well as a commitment to providing cutting edge tools and technology to artists, composers, and music aficionados.

In several creative fields, GANs have demonstrated incredible promise. In the context of developing music production and composition technology, applying GANs to music generation not only pushes the boundaries of AI but also advances the development of AI-driven music synthesis approaches.

The music industry is always changing due to shifting customer wants and trends. The music industry is in need of flexible and dynamic music production solutions, and this paper can assist meet that need. Examples of the industry's increasing needs include developing soundtracks for media, background music for games, and customized music for marketing campaigns.

Ultimately, the paper seeks to help democratize music production, foster creativity, and push the limits of what technology is capable of in the field of music. In an effort to push the boundaries of AI music composition while remaining approachable for novices, this research study investigates the creative potential of GAN-based AI in music generation. Our goal is to help novices in machine learning and music theory understand complex concepts by demystifying GAN architectures and their use in autonomous music composition. We provide a concise introduction to fundamental musical concepts to bridge technical aspects with artistic nuances. Additionally, our paper includes practical implementations and case studies to showcase the creative potential and limitations of GAN-based AI music generation systems.

Literature Review

GANs in Creative Domains

Semantic picture editing, style transfer, image synthesis, image super-resolution, and categorization are just a few of the many uses for GANs (Alqahtani et al., 2019). They have also shown notable increases in power density and efficiency when employed in power devices including battery chargers, high-voltage DC/DC converters, and front-end power factor correction. GANs have been very effective in the area of picture generation, finding use in video, image-to-image, text-to- image, and image synthesis

(Li et al., 2022). Moreover, GANs have been used in a wide range of fields, such as banking, marketing, fashion design, sports, music, protein engineering, astronomical data processing, remote sensing picture dehazing, and crystal structure synthesis (Dash et al., 2021).

In terms of creative domains, following are some examples of applications of GANs:

Generate Realistic Photographs

With applications in many different domains, GANs have demonstrated considerable potential in image production. (Chi et al., 2022) gives a thorough introduction to GANs in image production, emphasizing how they can increase classifier accuracy and data balance. (Yang et al., 2017) presents LR-GAN, a model that takes scene structure and context into account while creating realistic images. (Marchesi, 2017) offers a Deep Convolutional GAN-based optimized image generating technique that produces photorealistic, high-resolution images. (Meng & Guo, 2021) goes on to investigate the effectiveness of GAN models in the creation and categorization of images, concluding that GANs with convolution layers perform better than other models in these domains. Together, these researches highlight the potential of GANs in photo production, especially when it comes to enhancing the realism and quality of images.

Figure 1

Progress visualization of GANs from 2014 to 2017. The images, in order, are from (Goodfellow et al., 2014), (Radford et al., 2015), (Liu & Tuzel, 2016) and (Karras et al., 2018)

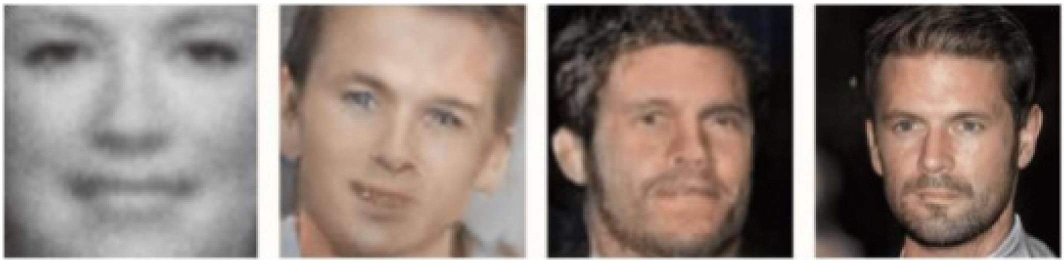
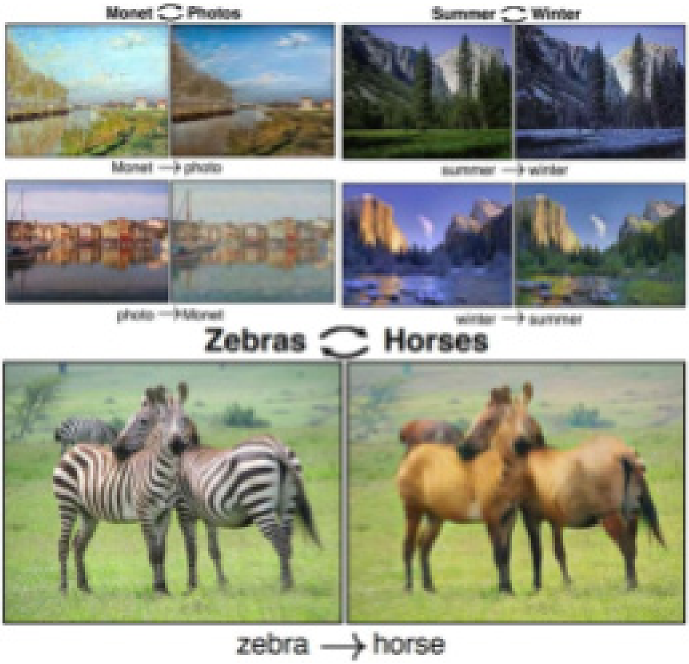


Image-to-Image Translations

New developments in GAN-based image-to-image translation have spawned the creation of multiple creative models. (Yu et al., 2019) introduced SingleGAN, a technique that achieves improved performance for multi-domain image translation by using a single generator. (Tang et al., 2022) presented SelectionGAN, a model for guided image-to-image translation problems that integrates semantic assistance to yield high-quality output. InstaGAN, introduced by (Mo et al., 2019), enhances multi-instance transfiguration by utilizing instance data and a context-preserving loss. (Lata et al., 2019) illustrated the significance of hyper-parameter adjustment on model performance by using Conditional GANs for image-to- image translation. Together, these works demonstrate the promise of GANs in picture translation as well as the variety of methods that may be applied to improve their efficiency.

Figure 2

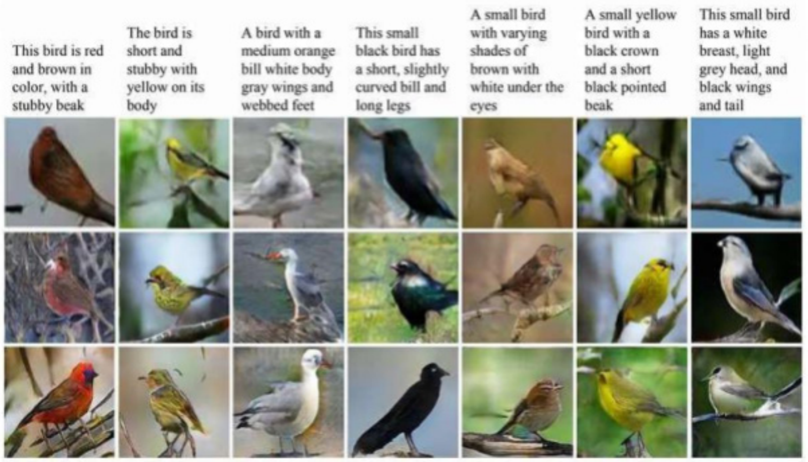
Examples of image-to-translation using GAN (Yu et al., 2019)



Text-to-Image Translation

Figure 3

Generated Images corresponding to the user input (Zhang et al., 2017)


This bird is red and brown in color, with a stubby beak
The bird is short and stubby with yellow on its body
A bird with a medium orange bill white body gray wings and webbed feet
This small black bird has a short, slightly curved bill and long legs
A small bird with varying shades of brown with white under the eyes
A small yellow bird with a black crown and a short black pointed beak
This small bird has a white breast, light grey head, and black wings and tail

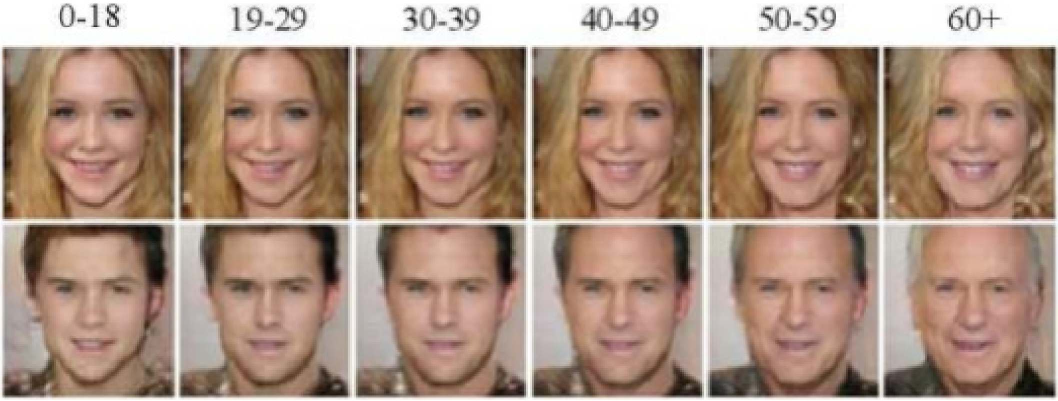
Many researchers explored the use of GANs for text-to- image translation, with (Perumalraja et al., 2022) specifically proposing a Stacked GAN approach for this task. In order to create high-resolution, photorealistic photographs, their method consists of two stages: the first stage produces low- resolution photos, and the second step refines them.

Face Aging

The application of GANs for facial aging has been investigated in a number of studies. Contextual GANs were introduced by (Liu et al., 2017) and take into account the transition patterns in aging faces. (Antipov et al., 2017) used an "Identity- Preserving" optimization technique to concentrate on maintaining the original identity in aged faces. (Liu et al., 2018) presented a DCGAN-based model that creates robust and lifelike face images by mapping a face to a latent vector and age-conditional vector. (Genovese et al., 2019) advanced this work by presenting a framework for understanding GANs, dissecting the model's internal organization, and demonstrating that general purpose picture datasets may be used to teach the aging transformation. Together, these researches show how GANs can be used to address face aging, with particular emphasis on identity preservation, transition pattern capture, and realistic image generation.

Figure 4

Face Aging according to age using GAN (Antipov et al., 2017)

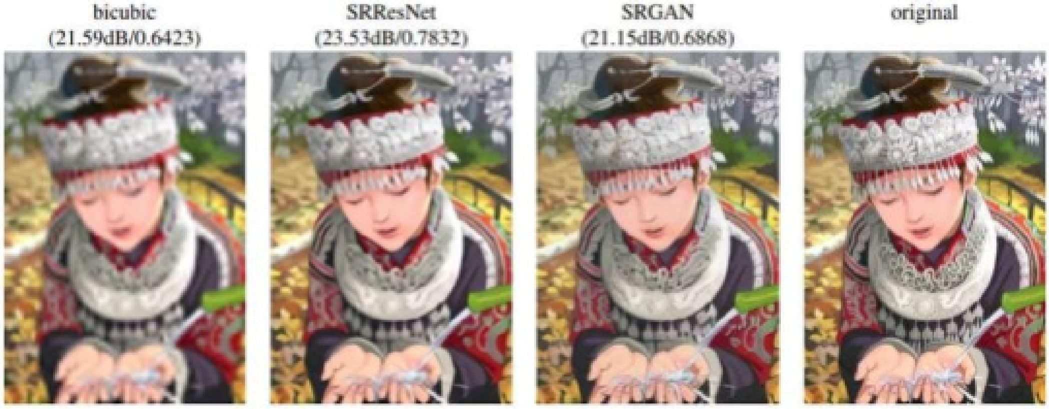


Super Resolution

The application of GANs for super resolution has been investigated in a number of research. Many discovered that GANs can improve picture details and create visually pleasing frames, while some research like (Gopan & Kumar, 2018) pointed out a somewhat lower PSNR (peak signal-to-noise ratio). (Liu et al., 2019) added a natural picture gradient prior and a style map from a visible image to further enhance GAN-based super resolution. Building on this work, (Xue et al., 2020) achieved outstanding results in image super resolution using a GAN model with residual blocks and a Wasserstein distance loss function. Last but not least, (Liu et al., 2019) showed how GANs might improve the resolution of coherent imaging systems, especially in lens-less on-chip holographic microscopes.

Figure 5

Increased Resolution of an Image using GAN (Ledig et al., 2017)



Systems for Automated Music Generation

Many computational approaches and techniques have been explored in the exploration of automated music generating systems to endow machines with musical creativity. A diverse blend of creativity and technology is reflected in the landscape of automated music generation systems, which range from rule-based algorithms to neural network structures.

Rule-Based Systems

Both (Goodman & Spangler, 1999) and (Friberg, 1991) created rule-based systems for generating music; former concentrated on transforming written compositions into performances, while latter concentrated on harmony and real-time accompaniment. (Bach, 2008) extended this work by using a Macintosh computer to implement the rules in Lisp. A more versatile and broad method was presented by (Lichtenwalter et al., 2009), who offered a system that generates rules from a training set of musical data using learning algorithms. Together, these investigations show the promise of rule-based systems for producing music, and Lichtenwalter's method presents a fruitful direction for further investigation.

Markov Model Systems

The application of Markov models in music generating systems has been the subject of several studies. Markov chains and hidden Markov models are used by De (Merwe & Schulze, 2011) and

(Thornton, 2009) to record and reproduce musical genres. In order to improve long-term coherence, (Herremans et al., 2015) expands on this work by putting out a strategy for enforcing structure and repetition in music. (Li et al., 2019) adopts a novel strategy and surpasses conventional approaches in music play sequence prediction by employing a mixed hidden Markov model. Together, these researches show how Markov models may be used to generate music for applications in sequence prediction and style replication.

Deep Learning Models

The transformer model and GANs have been the subject of recent research studying the application of deep learning in music production (Min et al., 2022). These models have demonstrated potential in deciphering linkages throughout lengthy musical sequences and acquiring knowledge of compositional conventions (Cheng et al., 2020). Deep learning for music creation does have several drawbacks, though, such as the inability to directly regulate production and the propensity to replicate the training set without genuine innovation (Briot & Pachet, 2017). Deep learning has been effectively used to generate music utilizing raw audio recordings and frequency domain data, despite these obstacles (Bhave et al., 2019).

Evolutionary Computational Systems

Evolutionary computing has been investigated in a number of research related to music generating systems. (Weale & Seitzer, 2003) concentrated on utilizing a genetic algorithm to generate contrapuntal music, whereas (Chen, 2007) created an interactive system that generates personalized music depending on user choices. (Wilson & Fazenda, 2016) introduced evolutionary computation to intelligent music composition, combining user feedback and domain knowledge. (Yiiksel et al., 2011) employed evolutionary algorithms and neural networks to autonomously compose music. Together, these experiments show how evolutionary computing may be used to produce a wide range of customized musical compositions.

Research Gap

The GANs architecture has been vastly used for image generation and manipulation, but very little work has been done with its utilization in music generation. GANs are generally used with the same architecture of models for both generator and discriminator. This study tries to bridge that gap and explore the usage of GANs in music generation with LSTM as generator and Patch GAN as discriminator.

Research Methodology

Music Properties and notations in GAN-Based Music Generation

It is essential to comprehend the basic notations and characteristics of music in order to interact with AI music production in an efficient manner. This section aims to simplify these ideas for both novices and experts in order to promote understanding and make it easier for people to get started with AI-powered music production. People can better manage the complexities of AI music generation if they understand the fundamentals of musical notation and the traits that go along with it. We hope that with this core understanding, users may investigate the nexus between artificial intelligence and music theory, enabling them to take advantage of technological breakthroughs in music production and composition.

Melody

In music, a series of notes and rests that are heard as a unified whole is referred to as a melody. It is the principal linear component that conveys a composition's central melodic concept or theme. Pitch, rhythm, and contour combine to form melody, which produces a recognizable and memorable series of sounds.

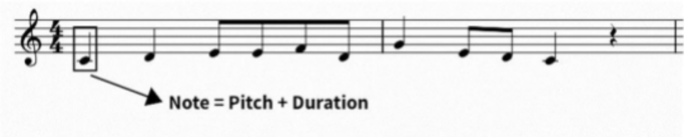
For novices, melody comprehension is figuring out how individual note pitches relate to one another and change over time. A melody is often made up of notes placed in a certain order to convey a feeling of musical direction and emotion.

In essence, melody is the heart of a musical piece, providing listeners with a focal point and emotional resonance. Learning to recognize, value, and compose melodies is essential for novices who want to get a better comprehension and bond with music.

Notes

Figure 6

Note of a melody

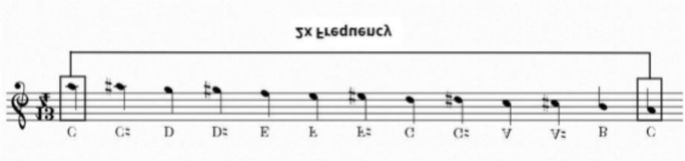


A note has two properties; pitch and duration. The pitch indicated how high or low the note sounds. It is directly correlated with the frequency of a sound i.e. the higher the pitch of a note, higher is the frequency of that sound. The duration of a note, which varies depending on the kind of note, is the amount of time it is played. In contemporary music, the whole note has the longest length.

Octave

Figure 7

An octave starting at C note

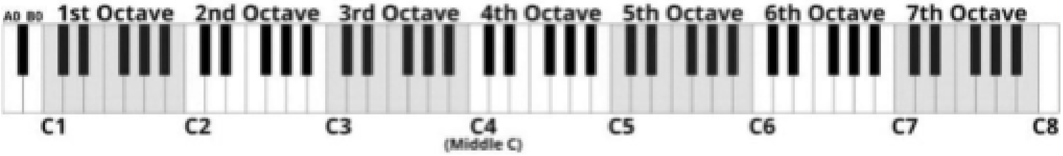


An octave, often known as a perfect octave, in music is a range of notes that fall between two notes, each of which vibrates at twice or half the frequency of the other. In the western musical scale, there are 12 notes in every octave.

Scientific Pitch Notation (SPN)

Figure 8

Full piano keyboard (all 88 keys) showing the different octaves [55]

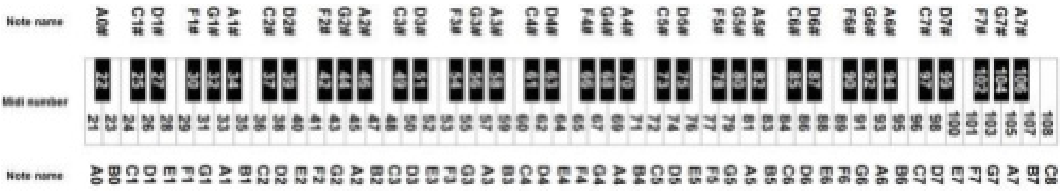


Scientific pitch notation (SPN), often referred to as international pitch notation (IPN) or American standard pitch notation (ASPN), is a technique for indicating musical pitch that combines a musical note name (with accidental if needed) and a number identifying the pitch's octave. For example, to denote a C note at 4th octave, it is denoted as C4. Similarly, D3 to denote a D note at 3rd octave. C4 is called the middle C, denoting the mid of the octave cale and the total available number of octaves depends on the instrument used to produce the sound. For example, the organ has one of the widest ranges of octave span, which is eleven.

MIDI Note Notation

Figure 9

88-notes classical keyboard-Note names and MIDI numbers [57]



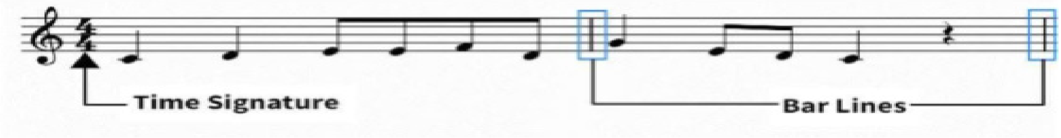
While it is convenient for humans to infer notes and their octaves using the Scientific Pitch Notation, the process can be made easier for computers by just using integers. MIDI is one among such notations generally used for digital music for playing, editing and recording music. In MIDI notation, the notes are mapped to numbers. According to MIDI notation, MIDI note 0 is assigned to C−1 (which is five octaves below C4 or Middle C and the lowest note on the two largest organs in the world; however, its overtones are audible), MIDI note 21 is assigned to A0 (the bottom key of an 88-key piano), MIDI note 60 is assigned to C4 (Middle C), and MIDI note 108 is assigned to C8 (the top key of an 88-key piano).

Note Values and Beats

A beat is the fundamental unit of time used to measure cycles in music theory. The beat is commonly described as the rhythm that a listener would tap their toes to when listening to music. Similarly, note values tell us how many beats a note gets, or how long the note will last. A note value indicates the relative duration of a note, using the texture or shape of the notehead (♫), the presence or absence of a stem, and the presence or absence of flags/beams/hooks/tails. The following figure shows how many beats every note is worth.

Figure 10

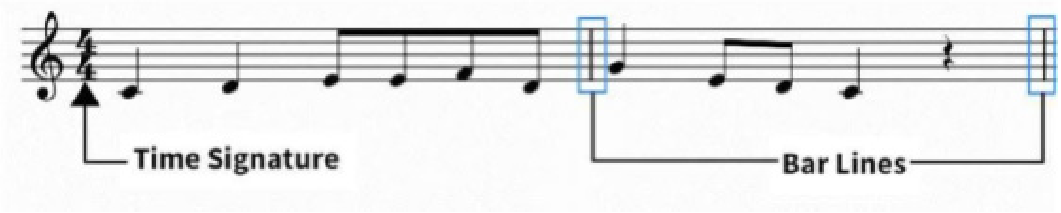
Relation between note values and beats [56]



Bar and Time Signature

Figure 11

Bar lines and Time Signature



Bar is a segment of music bounded by vertical lines, known as bar lines, usually indicating one of more recurring beats. Another important thing to learn in music theory is time signature in order to judge the output of the neural network. To check if the network is able to generate music in 4/4, 3/4, because different time signatures of the melody have different tone/accent/vibe. The time signature is written in fractional forms like 4/4, 3/4 etc. The numerator in these fractions denotes the number of beats there are in a bar and the denominator denotes the note value which is equal to one beat. The time signature 3/4 means that there are 3 beats in a bar and each beat is a quarter note.

Key

The key of a musical piece is the group of pitches or scales that form the center of musical composition in western classical music. A key consists of tonic and mode, e.g. C + Major = Cmaj

D + Minor = Dmin, where tonic being the note itself and mode being either major or minor. The major tone is usually construed as being happy sounding while the minor tone is sad sounding. With 12 notes and 2 modes, we altogether have 12 keys.

Transposition

The act of shifting a group of notes up or down in pitch by a consistent interval is referred to as transposition in music. This changes the key of the music, but the musical content remains the same.

GAN Architecture for Music Generation

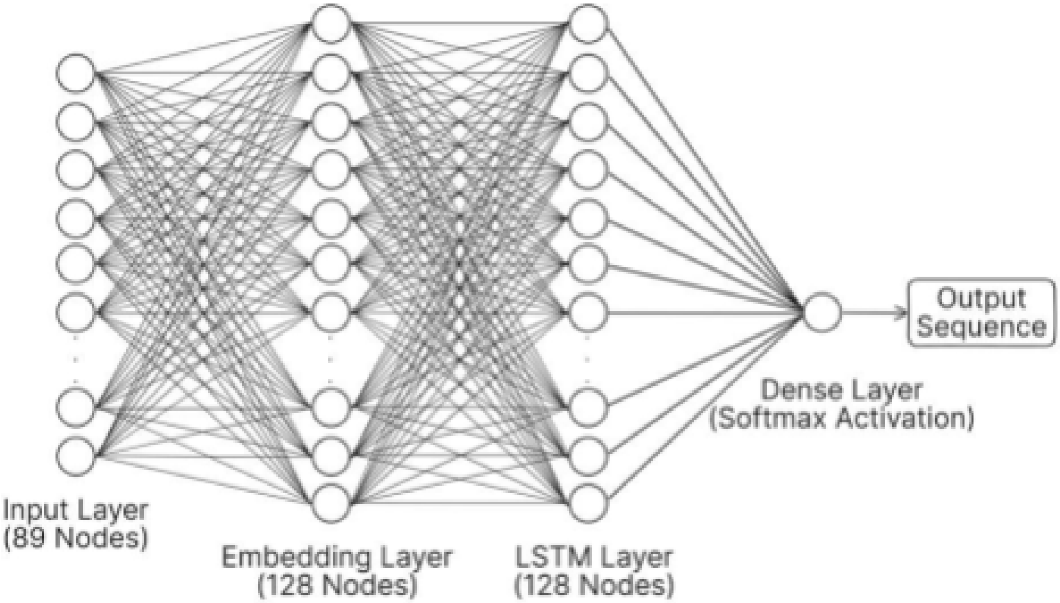
The GAN Architecture for Music Generation consists of LSTM Generator, Patch GAN Discriminator and a GAN framework that combines generator and discriminator into a single model for training.

LSTM Generator

The generator is a neural network with an LSTM foundation that creates musical sequences. It generates a series of music data from an input sequence. The generator is composed of an embedding layer, an LSTM layer, and a dense layer with softmax activation that is fully coupled. Visual representation of the generator model is given in figure 12.

Figure 12

LSTM Generator Model Diagram



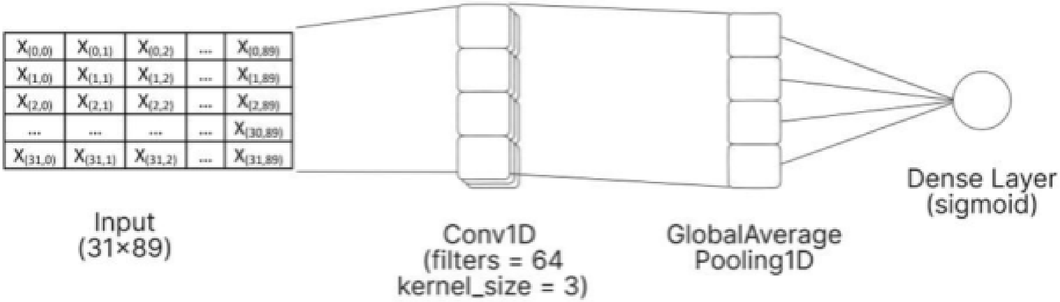
Input data is transformed into dense vectors of a predetermined size (latent dim) by the embedding layer. The embedded input sequence is processed by the LSTM layer, which also records temporal dependencies. A probability distribution across the set of potential musical events (such as pitches) is produced by the dense layer.

Patch GAN Discriminator

The discriminator distinguishes between artificially created and real music sequences using a patch-based discriminator. Visual representation of the discriminator model is given in figure 13.

Figure 13

Discriminator Model Diagram


$X_{(0,0)}$ $X_{(0,1)}$ $X_{(0,2)}$ ... $X_{(0,89)}$
$X_{(1,0)}$ $X_{(1,1)}$ $X_{(1,2)}$ ... $X_{(1,89)}$
$X_{(2,0)}$ $X_{(2,1)}$ $X_{(2,2)}$ ... $X_{(2,89)}$
... ... $X_{(30,89)}$
$X_{(31,0)}$ $X_{(31,1)}$ $X_{(31,2)}$ ... $X_{(31,89)}$
Input
(31×89)
Conv1D
(filters = 64
kernel_size = 3)
GlobalAverage
Pooling1D
Dense Layer
(sigmoid)

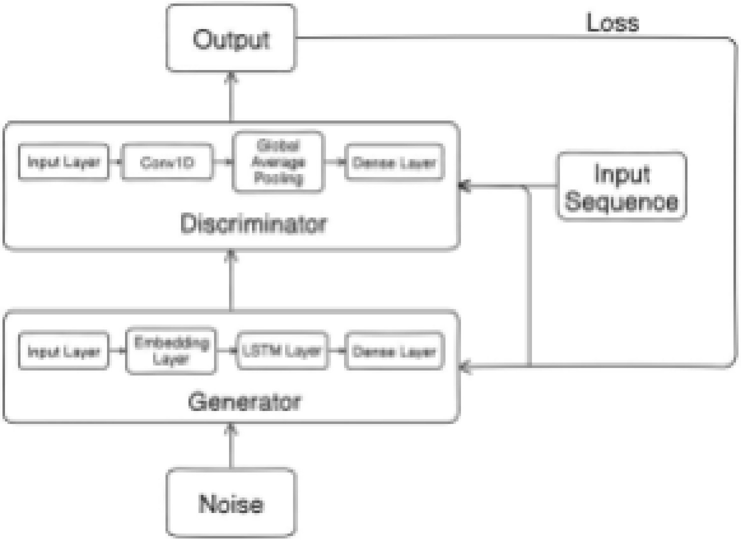
It receives input in the form of musical sequences and produces a probability reflecting the authenticity of the input. The discriminator uses a dense layer with sigmoid activation and a convolutional neural network (Conv1D) with leaky ReLU activation followed by global average pooling. The local patterns of input music sequences are captured by the convolutional layers. The spatial dimensions of the features are shrunk by the global average pooling layer. The probability that the input sequence is real is shown by the single output that the dense layer generates.

GAN Class

The generator and discriminator are combined into a single training model by the GAN class. In training, the discriminator's goal is to properly discriminate between genuine and produced sequences, while the generator attempts to create music sequences that are realistic enough to trick the discriminator. Both the discriminator and the generator are trained adversarially: the discriminator's loss is minimized to enhance its capacity to discern between created and genuine music, while the generator's loss is minimized to produce more realistic music. Visual representation of the whole GAN class is shown in figure 14.

Figure 14

GAN Model Diagram



Dataset

Maestro MIDI dataset was utilized to train the models since it provides an attractive resource for training music creation models, the. With more than 200 hours of recordings from the International Piano-e-Competition, MAESTRO offers a large body of work covering a wide range of musical styles and levels of difficulty. Models are able to learn from a wide range of musical patterns and subtleties because of this large and diverse dataset. The dataset captures both the rich sound and the specific note information with high temporal resolution (~3ms) and pairs it with well-matched MIDI data in CD-quality audio. Models can efficiently learn the complex link between musical notation and its accompanying sound because of this exact alignment, which is essential for producing musically cohesive compositions.

Every composition in MAESTRO has an annotation indicating the author, title, and performance year. Because of the comprehensive metadata available, models may be able to produce music that sounds realistic and include contextual information into the process, which might result in the development of compositions with stylistic and structural traits consistent with the information supplied. This also makes MAESTRO very suitable for the future work on the project, more of which is discussed in the later section of the paper.

Data Pre-processing and training

The goal of the data preprocessing part is to get note sequence and target data from the pitch information in the MIDI files in the form of tensors. To achieve this, from a list of MIDI files, each file is iterated and is passed to the preprocess_midi function. This function reads a MIDI file, extracts note information, and quantize time steps and returns a list of dictionaries, where each dictionary contains the following keys:

- pitch: The MIDI pitch of the note.
- velocity: The velocity of the note (loudness).
- start: The starting time step of the note.
- end: The ending time step of the note.

The result from the function is stored in a list of notes. From the list, another list of valid notes is formed, which is a list of notes that have a pitch value from 0 to 88.

The list of valid notes and sequence length (64 in this implementation) were passed to the prepare_sequences function. Because the list was too big for the system memory of the particular computer used to process the data, a batch size was also passed as parameter to the function and a helper function, prepare_sequence_helper was used in order to process the sequences in batches. With the help of the helper function, the prepare_sequence function returns sequences and targets, where sequences is a list of tensors, where each tensor represents a sequence of notes and targets is a list of tensors, where each tensor represents the target sequence (shifted by 1 step).

After the music data is processed, the generator is provided with noise data with similar dimensions as the processed data. Out of this noise data, the generator tries to produce some results. This result is evaluated by the discriminator. Binary_crossentropy function computes the loss, which indicates how different the generated result is from the actual music. The generator makes changes and tries to bring the result of loss function closer to 1. The weights of the two models are optimized by Adam optimizer.

Results and Discussion

Results

The implementation for the proposed system was done on a basic level and consisted broadly of four steps: data preprocessing, model architecture, model training and music generation. The following section covers each step in brief detail.

Model Architectures

As mentioned in the previous section, the model architecture consists of three components, a LSTM Generator, a discriminator and a GAN class that combines the generator and discriminator enabling them to train adversarially.

The LSTM Generator is a sequence-to-sequence model that makes use of an LSTM architecture to produce music. Three layers were implemented for the generator: embedding layer, LSTM layer and a dense layer.

Integer input representing musical pitches is sent into the Embedding layer, along with a unique token for padding. This layer uses an embedding matrix to convert each integer to a dense vector of size latent_dim (128). In this way, internal representations for every pitch may be learned by the model.

The LSTM layer then processes the sequence of embedded pitches. The model uses a single LSTM layer with units

(256) memory cells. The return_sequences=True argument ensures the output retains the sequence structure, allowing the network to model long-term dependencies between pitches.

The output of the LSTM layer is projected by the Dense layer onto a vector of size NUM_PITCHES + 1 (88+1). The additional unit compensates for the input's padding token. The output layer then applies a softmax activation function to transform the logits into probabilities, which indicate the possibility of each potential pitch at the following step in the musical sequence.

The discriminator used is a patch-based discriminator which focuses on local patches instead of entire sequences for more fine-grained analysis. It uses few layers for efficiency and faster training. The activation functions that are used, Leaky ReLU and sigmoid, promote gradient flow and provide a probabilistic interpretation of output respectively.

The discriminator consists of 4 layers; input layer, convolutional layer, global average pooling layer and dense layer.

Music sequences are accepted as input by the input layer, which anticipates a shape of [batch_size, sequence_length, features]. For consistency, an additional batch dimension is introduced if just one sequence (shape [31, 89]) is given. Features are extracted by the convolutional layer from certain regions within the music stream. With a given kernel_size, it applies 1-dimensional filters. "Same" padding is used to maintain input length and for non- linearity, leaky ReLU activation is used.

By averaging throughout the full sequence, the global average pooling layer compresses feature maps.

It reduces dimensionality while capturing global information.

Discriminating decision of the model is represented by a single output value that is projected from pooled features by the dense layer. For output constraint, it employs a sigmoid activation function between 0 (fake) and 1 (real). The MusicGAN class creates a GAN model using TensorFlow's Keras API and houses the generator and discriminator networks. The initialization function of the class initializes the model with the provided generator and discriminator. The class also has a compile function which configures the model's compilation, specifying loss function and optimizer for training. The train_step function of the MusicGAN class implements a single training step, handling both generator and discriminator training. The generate function of the class generates new music sequences using the trained generator model.

Training

The primary container for the whole GAN system is the MusicGAN class, which derives from tf.keras.Model. It has two essential submodels: Generator and Discriminator.

The compile function sets up the GAN model's necessary parameters prior to training. It clarifies two important components:

- Loss Function: This function calculates the penalty for the model's predictions deviating from the desired outcome. The model makes use of the binary_ crossentropy function in this instance. Tasks requiring binary classifications, such as determining if a musical sequence is authentic or fraudulent, are a good fit for this function.
- Optimizer: Based on the determined loss values, this method is essential in adjusting the internal parameters (weights) of the model. The given code makes use of the Adam optimizer, a popular and effective neural network training tool.

Within the GAN training process, the train_step method captures the fundamental reasoning of a single training iteration. This process may be distilled into a few essential steps:

- Data Preparation: To provide real music sequences for the generator and discriminator to refer to, the model extracts them from the training set. The model also produces random noise to provide the generator ideas for where to begin when creating musical sequences.
- Generator Training: In an effort to replicate the style and organization seen in the actual music data, the generator uses the noise that has been supplied to produce its own musical sequences. Subsequently, the model employs the binary_crossentropy function to compute a loss (gen_loss). This loss measures the difference between the targeted result (ones, representing authentic music) and the discriminator's predictions for the generated sequences (preferably near ones, representing actual music).
- Discriminator Training: Both the created and actual music sequences are evaluated independently by the discriminator. For every sequence, it produces a probability score that represents how likely it is to be true. Disc_loss is another loss that is computed with the binary_crossentropy function. This loss represents the discrepancy between the genuine labels and the discriminator's actual predictions, which are zeros for bogus sequences and ones for authentic sequences.
- Back-propagation and Optimization: The generator and discriminator losses' gradients (rates of change) with respect to their individual internal parameters (weights) are computed by the model via a process known as backpropagation. These gradients are used by the selected optimizer, Adam in this instance, to update the weights of the two models. With this version, the discriminator's capacity to

distinguish between actual and produced sequences will be improved while the generator's ability to produce realistic music that fools the discriminator will also be improved.

- Loss Reporting: The model reports the generator and discriminator losses after every training phase. These losses serve as vital indicators for tracking the entire training process' development and assessing its efficacy.

The real training starts when the model is set up and each training step is specified. This is made easier by the tf.keras.Model class's fit function. For a predetermined number of epochs (full runs of the dataset), this technique iterates over the training dataset. The train_step technique is run several times in each epoch, progressively fine- tuning the discriminator and generator in an adversarial fashion until the generator reaches its goal of producing music that is imperceptible to the astute discriminator as genuine music.

Discussion

Implications for the Industries

Customizable music generating systems might have a big influence on a lot of different industries:

1. Music Industry
    - Composition assistance: Using the technique, songwriters and composers can get beyond writer's block, come up with new takes on well-known subjects, or try out novel approaches and genres.
    - Production efficiency: Using the technology, producers may swiftly produce backing tracks, adapt songs for different audiences, or customize songs for particular listeners.
2. Entertainment Industry
    - Film and video game scoring: Composers can save time and resources by having the system quickly produce background music that suits the mood and concept of a scene.
    - Personalized soundtracks: Custom soundtracks that adjust to the audience or surroundings may be made for museums, amusement parks, and other types of venues.
    - Interactive experiences: AI-generated music may be used to create dynamic, immersive soundtracks for games and virtual reality experiences that react to player activities.
3. Education and Research
    - Music education: If students give the system some beginning points and explore other options, they can experiment with writing and arranging music.
    - Music therapy: By employing the technology to customize music to each

patient's unique requirements and tastes, therapists may make music treatments more unique for their patients.
- Musicology: By examining the system's produced outputs, researchers may learn about composing methods, musical theory, and the development of musical styles.

4. General Public
   - Accessibility and creativity: The method encourages creativity and self-expression by enabling anybody to compose music, regardless of experience or aptitude.
   - Personalization and entertainment: Users may create personalized music for activities like meditation, relaxation, or just to enjoy the novelty of AI-generated music.

Challenges and Future Directions

A unique set of challenges arises while training Generative Adversarial Networks (GANs) for music production, especially when restricted computing resources are available. A major challenge is determining the ideal hyperparameters for the model's performance.

Because GANs are infamously sensitive to hyperparameter settings, precise modifications are necessary to get desired results, such as realistic music creation. This procedure becomes more laborious in contexts with limited resources since it might be computationally costly to explore a wide variety of hyperparameter combinations.

In addition, when there are little resources available, the training procedure itself presents difficulties. To manage the intricate computations needed to produce realistic music, training GANs frequently demands a large processing power. A lack of resources might cause training times to increase, which can impede the process of development and perhaps lower the overall quality of created music.

Generative Adversarial Networks, or GANs, are becoming a very useful tool for creating music. Two major issues with the paradigm of model in this paper, though, are that there is little variation in the outputs that are produced and there is little control over the musical style. In order to overcome these constraints, this research suggests two next directions:

1. Enhancing Dataset Diversity for Richer Outputs: For GANs to produce a variety of outputs, they must be trained on a larger spectrum of musical genres and styles. A wider range of musical genres, including jazz, world music, folk, and experimental music, should be included in the training set to give the model a deeper comprehension of musical structures and patterns. The generator network can now create outputs that are more than just variations on a single theme thanks

to this increased exposure, which enables it to catch the subtleties of many musical styles. Including a variety of musical components within a genre might further add to the model's knowledge base. By including changes in tempo, instrumentation, and composing methods, the GAN may produce music that accurately captures the complexity of a certain genre.

2. Leveraging Keyword Labels for Style-Based Generation:The capacity to make music based on style keywords would be a big development, even if the existing model can generate music simply feeding it starting notes. By adding keyword labels to the training dataset, this may be accomplished. These designations could include musical styles (like "jazz," "heavy metal"), emotional states (like "upbeat," "melancholy"), or even particular instruments (like "flute solo"). The connections between these keywords and the associated musical elements might then be taught to the model. The user might enter a list of keywords and, if desired, a brief musical seed throughout the generating process. The generating process would then be guided by the information provided by the GAN, resulting in music that is coherent and musically pleasing while adhering to the designated style. This would provide consumers more creative freedom and allow them to customize the generated music to suit their own requirements and aesthetic preferences.

Conclusion

The potential of Generative Adversarial Networks (GANs) for music production was investigated in this study. After

50 training epochs, the implemented model had an astounding accuracy of 91.82%, indicating its efficacy in picking up the subtleties of musical data.

In addition to its remarkable accuracy rating, the produced music had a high level of coherence. This shows that the model can not only understand the basic principles of music theory but also pick up on the minute details that add to a piece's emotional resonance and distinctive style. This ability of GANs to replicate these details shows how they can produce music that is both technically accurate and artistically appealing to listeners.

All things considered; this study's results show how much GANs may be used to advance the field of music creation. GANs have the potential to produce high-fidelity, stylistically varied music that can enthrall listeners and open up fresh creative opportunities in the music industry with more research and development.

References

AlDahoul, N., Hong, J., Varvello, M., & Zaki, Y. (2023, 10 26). *Exploring the potential of generative AI for the World Wide Web*. arXiv. https://doi.org/10.48550/arXiv.2310.17370

Alqahtani, H., Thorne, M. K., & Kumar, G. (2019, 12 19). Applications of generative adversarial networks (GANs): an updated review. *Archives of Computational Methods in Engineering, 28*(2), 525-552. https://doi.org/10.1007/s11831-019-09388-y

Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). Face aging with conditional generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP) (pp. 2089-2093)*. IEEE. https://doi.org/10.1109/ICIP.2017.8296650

Bach, J. (2008). *Generative rules for music performance: a formal description of a rule system*. Retrieved from http://www.speech.kth.se/prod/publications/files/1771.pdf

Bhave, A., Sharma, M., & Janghel, R. R. (2019). *Music generation using deep learning. In Soft Computing and Signal Processing (pp. 203-211)*. Springer Singapore. https://doi.org/10.1007/978-981-13-3393-4_21

Briot, J. P., & Pachet, F. (2017, 12 09). Deep learning for music generation: challenges and directions. *Neural Computing and Applications, 32*, 981-993. https://doi.org/10.1007/s00521-018-3813-6

Cao, Y., Li, S., Liu, Y., Yan, Z., Dai, Y., Yu, P. S., & Sun, L. (2023, 3 7). *A comprehensive survey of AI-generated content (AIGC): a history of generative AI from GAN to ChatGPT*. arXiv. https://doi.org/10.48550/arXiv.2303.04226

Chen, Y. p. (2007, 02). *Interactive music composition with evolutionary computation*. Retrieved from https://api.semanticscholar.org/CorpusID:17238192

Cheng, P. S., Lai, C. Y., Chang, C. C., Chiou, S. F., & Yang, Y. C. (2020). A variant model of TGAN for music generation. In *Proceedings of the 2020 Asia Service Sciences and Software Engineering Conference (pp. 40–45)*. Association for Computing Machinery. https://doi.org/10.1145/3399871.3399888

Chi, W., Choo, Y. H., & Goh, O. S. (2022, 01). Review of generative adversarial networks in image generation. *Journal of Advanced Computational Intelligence and Intelligent Informatics, 26*(1), 3-7. https://doi.org/10.20965/jaciii.2022.p0003

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2017, 10 19). *Generative adversarial networks: an overview. IEEE Signal Processing Magazine, 35*, 53-65. https://doi.org/10.1109/MSP.2017.2765202

Dash, A., Ye, J., & Wang, G. (2021, 10 01). A review of generative adversarial networks (GANs) and its applications in a wide variety of disciplines: from medical to remote sensing. *IEEE Access, 12*, 18330-18357. https://doi.org/10.1109/

ACCESS.2023.3346273

Epstein, Z., & Hertzmann, A. (2023, 06 15). *Art and the science of generative AI. Science, 380*(6650), 1110-1111. https://doi.org/10.1126/science.adh445

Friberg, A. (1991, 07). Generative rules for music performance: a formal description of a rule system. *Computer Music Journal, 15*, 56-71. https://doi.org/10.2307/3680917

Genovese, A., Piuri, V., & Scotti, F. (2019). Towards explainable face aging with generative adversarial networks. In *2019 IEEE International Conference on Image Processing (ICIP) (3806-3810).* IEEE. https://doi.org/10.1109/ICIP.2019.8803616

GM, H., Gourisaria, M. K., Pandey, M., & Rautaray, S. S. (2020, 11). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review, 38*, 100285. https://doi.org/10.1016/j.cosrev.2020.100285

Goodfellow, I., Abadie, J. P., Mirza, M., Xu, B., Farley, D. W., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems (Vol. 27)*. Curran Associates, Inc.

Goodfellow, I., Abadie, J. P., Mirza, M., Xu, B., Farley, D. W., Ozair, S., Courville, A., & Bengio, Y. (2020). *Generative adversarial networks. In Communications of the ACM (Vol. 63, pp. 139-144).* https://doi.org/10.1145/3422622

Goodman, R., & Spangler, R. R. (1999). *Rule-based analysis and generation of music.* Retrieved from https://doi.org/10.7907/YXTQ-4057

Gopan, K., & Kumar, G. S. (2018). Video super resolution with generative adversarial network. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1489-1493)*. IEEE. https://doi.org/10.1109/ICOEI.2018.8553719

Herremans, D., Weisser, S., Sörensen, K., & Conklin, D. (2015, 11 30). Generating structured music for bagana using quality metrics based on Markov models. *Expert Systems with Applications, 42*(21), 7424-7435. https://doi.org/10.1016/j.eswa.2015.05.043

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1710.10196

Lata, K., Dave, M., & Nishanth, K. N. (2019). Image-to-image translation using generative adversarial network. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 186-189)*. IEEE. https://doi.org/10.1109/ICECA.2019.8822195

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 105-114)*. IEEE. https://doi.org/10.1109/CVPR.2017.19

Li, J., Li, T., Lin, R., & Nie, Q. (2022, 09 23). GAN-based models and applications. *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, 848-852. https://doi.org/10.1109/ICISCAE55891.2022.9927647

Li, T., Choi, M., Fu, K., & Lin, L. (2019). Music Sequence Prediction with Mixture Hidden Markov Models. In *2019 IEEE International Conference on Big Data (Big Data) (pp. 6128-6132)*. IEEE. https://doi.org/10.1109/BigData47090.2019.9005695

Lichtenwalter, R., Lichtenwalter, K., & Chawla, N. (2009). Applying learning algorithms to music generation. In *Indian International Conference on Artificial Intelligence*. Retrieved from https://www.researchgate.net/publication/220888519_Applying_Learning_Algorithms_to_Music_Generation

Liu, M. Y., & Tuzel, O. (2016). Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (Vol. 29)*. Curran Associates, Inc. https://doi.org/10.48550/arXiv.1606.07536

Liu, S., Sun, Y., Zhu, D., Bao, R., Wang, W., Shu, X., & Yan, S. (2017, 10 19). *Face aging with contextual generative adversarial nets. Association for Computing Machinery*. https://doi.org/10.1145/3123266.3123431

Liu, S., Yang, Y., Li, Q., Feng, H., Xu, Z., Chen, Y., & Liu, L. (2019). Infrared image super resolution using GAN With infrared image prior. In *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)* (pp. 1004-1009). IEEE. https://doi.org/10.1109/SIPROCESS.2019.8868566

Liu, T., Haan, K. d., Rivenson, Y., Wei, Z., Zeng, X., Zhang, Y., & Ozcan, A. (2019, 03 08). Deep learning-based super-resolution in coherent imaging systems. *Scientific Reports, 9*(1), 3926. https://doi.org/10.1038/s41598-019-40554-1

Liu, X., Xie, C., Kuang, H., & Ma, X. (2018, 02 01). Face aging simulation with deep convolutional generative adversarial networks. *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. https://doi.org/10.1109/ICMTMA.2018.00060

Marchesi, M. (2017, 05 31). *Megapixel size image creation using generative adversarial* *networks*. arXiv. https://doi.org/10.48550/arXiv.1706.00082

Meng, H., & Guo, F. (2021). Image classification and generation based on GAN model. In *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 180-183)*. IEEE Computer Society. https://doi.org/10.1109/MLBDBI54094.2021.00042

Merwe, B. v. d., & Schulze, W. (2011, 07 01). Music generation with Markov models. *IEEE MultiMedia, 18*, 78-85. https://doi.org/10.1109/MMUL.2010.44

Min, J., Liu, Z., Wang, L., Li, D., Zhang, M., & Huang, Y. (2022, 11 27). Music generation system for adversarial training based on deep learning. *Processes, 10*(12). https://doi.org/10.3390/pr10122515

Mo, S., Cho, M., & Shin, J. (2019, 01 02). *InstaGAN: instance-aware image-to-image translatio*n. arXiv. https://doi.org/10.48550/arXiv.1812.10889

Perumalraja, R., Arjunkumar, A. S., Mohamed, N. N., Siva, E., & Kamalesh, S. (2022). Text to image translation using GAN with NLP and computer vision. *Periodico*. https://doi.org/10.37896/pd91.4%2F91449

Radford, A., Metz, L., & Chintala, S. (2015, 11). *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv. https://doi.org/10.48550/arXiv.1511.06434

Tang, H., Torr, P. H.S., & Sebe, N. (2022, 10 10). Multi-channel attention selection GANs for guided image-to-image translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(5), 6055-6071. https://doi.org/10.1109/TPAMI.2022.3212915

Thornton, C. J. (2009). *Hierarchical Markov modeling for generative music. International Conference on Mathematics and Computing*. Retrieved from https://www.semanticscholar.org/paper/Hierarchical-Markov-Modeling-for-Generative-Music-Thornton/e93831fbd86b7c193c62922646bb1e27b38cd9fe

Weale, T., & Seitzer, J. (2003). EVOC: a music generating system using genetic algorithms. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence (pp. 1383 - 1384)*. Morgan Kaufmann Publishers Inc.

Wilson, A., & Fazenda, B. (2016, 09). *An evolutionary computation approach to intelligent music production informed by experimentally gathered domain knowledge*. London, UK. https://www.researchgate.net/publication/319881165_An_Evolutionary_Computation_Approach_to_Intelligent_Music_Production_informed_by_Experimentally_Gathered_Domain_Knowledge

Xue, X., Zhang, X., Li, H., & Wang, W. (2020). Research on GAN-based image super-resolution method. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 602-605)*. IEEE. https://doi.org/10.1109/ICAICA50127.2020.9182617

Yang, J., Kannan, A., Batra, D., & Parikh, D. (2017, 08 02). *LR-GAN: layered recursive generative adversarial networks for image generation*. arXiv. https://doi.org/10.48550/arXiv.1703.01560

Yiiksel, A. Ç., Karci, M. M., & Uyar, A. Ş. (2011). Automatic music generation using evolutionary algorithms and neural networks. In *2011 International Symposium on Innovations in Intelligent Systems and Applications (pp. 354-358*). IEEE. https://doi.org/10.1109/INISTA.2011.5946091

Yu, X., Cai, X., Ying, Z., Li, T. H., & Li, G. (2019). *SingleGAN: image-to-image translation by a single-generator network using multiple generative adversarial learning. In Computer Vision – ACCV (pp. 341-356)*. https://doi.org/10.1007/978-3-030-20873-8_22

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxa, D. (2017). StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV) (pp. 5908-5916)*. IEEE. https://doi.org/10.1109/ICCV.2017.629

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV) (pp. 2242-2251)*. IEEE. https://doi.org/10.1109/ICCV.2017.244