Intelligent System Based on Deep Learning Technique for Accompaniment Music Generation

Nermin Naguib J. Siphocly[1a], El-Sayed M. El-Horbaty[1b] and Abdel-Badeeh M. Salem[1c]

Abstract – **Utilizing modern deep learning techniques in music generation is currently a hot research topic. The objective of this paper is to develop an intelligent accompaniment music generator using pix2pix GAN (generative adversarial network) deep networks. We present a novel representation of musical data into images through color encodings. We compare the effect of our suggested data representation technique on the pix2pix network learning as opposed to the traditional binary encoding scheme used in the earlier literature. Our experimental results show that our music representation achieved better results on pix2pix GANs over the traditional representations; reaching a loss function value of 0.001.**

Keywords – **Music Generation, Artificial Intelligence, Deep Learning.**

I. Introduction

In the past few years, computers have been excessively used in aiding music composers in composing new musical pieces quickly and adjectively. Computers can presently even generate music that have a certain style or genre specified by the user [1]. Moreover, computer music composition has opened the door for non-professional music amateurs to compose highly acceptable musical pieces. Artificial intelligence (AI) is one of the main reasons for making all this possible. AI techniques were utilized in the field of computer music composition for performing various composition tasks such as: main melody composition [2], accompaniment music generation [3], and bassline formation [4].

Deep learning is a field of artificial intelligence that studies machine learning through deep neural networks. There are various techniques of deep learning; nonetheless, the most popular and powerful is the recently developed *generative adversarial networks (GANs)* [5].

In this paper we are concerned with accompaniment music generation using the deep learning network "pix2pix GANs" [6]. Our objective is to develop an application that takes as an input a main melody and generates accompaniment music corresponding to it.

The rest of this paper is organized as follows Section II gives a theoretical background about the technique used in our system; "pix2pix GANs", in addition to shedding the light on the state-of-the-art in the field. Section III describes our methodology. Section IV illustrates our experimental results pointing out to our implementation details. Section V discusses the results. Finally, Section VI gives final thoughts and conclusions to our work marking out the future research opportunities.

II. Background and Related Work

A. Pix2pix GANs Deep Learning Networks

In traditional GAN generative models, the network learning is achieved by finding a mapping from a latent vector *z* of random noise and an output image *y*, G : z → *y*. However, conditional GANs were later developed to learn a mapping from random noise vector *z* in addition to an observed image *x* to output image *y*, G : {*x*, *z*} → *y*. One of the tasks of the generator G is to produce images that look so "real", so it is trained to fool the discriminator D, which is an adversarially trained to distinguish the generator's "fakes" from real images. The objective function of a conditional GAN is:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1-D(x,G(x,z)))], \quad (1)$$

where D tries to maximize this objective against G which tries to minimize it, i.e. $G^* = \arg\min_G \max_D L_{cGAN} (G, D)$.

The second task of the generator is to be close to the ground truth output image, the thing that can be achieved using L1 distance (binary cross entropy) as follows:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]. \quad (2)$$

Pix2pix final objective is:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G). \quad (3)$$

B. Related Work

From the most famous and earliest research that utilized GANs in music generation is that of Dong et al. [7] who arrived to develop a multitrack music generation system. They were from the first researchers to suggest training GANs on piano-rolls music representation encoding music as binary-valued time-pitch matrices. To overcome the problem of real value outputs from GANs (music notes must have integer values), they proposed an enhancement to their work in [8] by espousing binary neurons. They applied a refiner network to the output of the generator as a post-processing. The output layer of the introduced network is composed of binary neurons. Oza et al. [9] extended on Dong's work [8] by applying progressive training where pretrained, already converged, network layers are being continuously scaled by adding new layers to them.

To our knowledge, pix2pix GANs have been mainly used in music transcription rather than music generation. Music transcription is the process of generating musical scores from audio music, such as Kim et al. [10] who work on spectrograms

[1] Authors are with Faculty of Computer and Information Sciences, Ain Shams University, Cairo Egypt.
[a] E-mail: nermine.naguib@cis.asu.edu.eg
[b] E-mail: shorbaty@cis.asu.edu.eg
[c] E-mail: absalem@cis.asu.edu.eg

of audio files. In this work we use pix2pix GANs to generate accompaniment music given an input main melody.

III. Methodology

In this section we describe our system for accompaniment music generation using pix2pix GANs. We start by discussing our data representation technique, then we describe the system architecture.

A. Data Representation

As previously mentioned, pix2pix GANs performs style transfer from an image to another, hence, we convert musical data into images for the network to train on. The input to our network is image pairs for the main melody and its accompaniment respectively. The width of each image represents time steps, such that one second of time from the musical piece is represented in eight columns. The height of the image represents musical notes, such that each image has 106 rows corresponding to full keyboard notes (from A0 to G9).

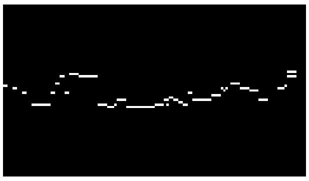

Fig. 1. Sample of main melody represented as an image

The main melody for each musical piece is represented as a black and white (binary piano-roll) image. At a given time step, each musical note in the melody is represented as a white pixel in its corresponding note position. Fig. 1 shows a sample image of the main melody representation.

We present a novel approach for representing the accompaniment music into images as opposed to the b/w pixels representation stated above. We encode the musical notes of the accompaniment music as colors inside the main melody. To perform this representation, we reserve 106 colors for representing each of the 106 musical notes. We start adding color values on top of the main melody black and white image in the following scheme;

- As a first step, we transform the accompaniment music into a black and white image in the same way as the main melody images were formulated.
- We scan the images generated from the previous step per time steps and for each time step we get all the concurrently played musical notes (i.e. all the white pixels at a given column), then, we lookup the corresponding colors for each of these notes.
- In parallel, we scan the main melody image and whenever we encounter a white pixel, we fill its RGB channels consequently with the color values of the corresponding accompaniment music notes values retrieved in the previous step.
- In some cases, we might need to color more than one pixel if there are more than three accompaniment musical notes played at the same time. If this is the case, we color the pixel below.

A sample image for the explained color encoding is shown in Fig. 2, magnified to plainly show the color variations. This is the color encoding of the accompaniment music that corresponds to the main melody displayed in Fig. 1.

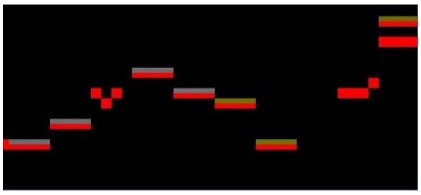

Fig. 2. A magnified sample of our color encoded accompaniment music

B. Network Architecture

Our pix2pix GAN network is, as standard, composed of a generator network and a discriminator network. The input to the generator is the main melody b/w images. As follows we describe the architecture of each of the generator and the discriminator networks.

The Generator

The generator takes as an input the main melody image and trains to generate a fake accompaniment image that corresponds to the input melody. Fig. 3. describes the generator's architecture where the input passes through a sequence of downsampling convolution layers followed by another list of upsampling convolution layers. The middle convolution layers are followed by batch normalization layers. The final upsampling convolution layer at the bottom defines the output accompaniment image.

The Discriminator

As in the traditional GANs, the discriminator's job is to judge whether the image generated from the generator network looks fake or real. However, in case of the pix2pix GANs, the discriminator takes an extra input; in addition to the image generated by the generator, the discriminator takes the main melody image too.

Fig. 4. shows the architecture of the discriminator network which takes as an input the fake accompaniment image (generated by the generator) "concatenated" with the main melody image (that was the input to the generator network). This input is passed through a list of convolution layers with leaky relu activation, followed by batch normalization layers. Each convolution layer downsamples the input by half. The final layer is a binary classification layer from 0 to 1 (from fake to real).

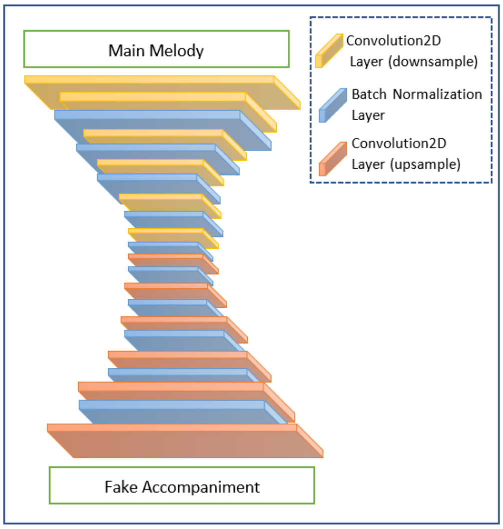
Main Melody
Convolution2D Layer (downsample)
Batch Normalization Layer
Convolution2D Layer (upsample)
Fake Accompaniment

Fig. 3. Architecture of the generator network

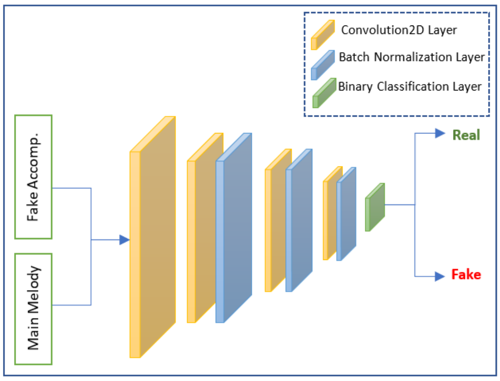

Fig. 4. Architecture of the discriminator network

IV. Experimental Tests and Results

We implemented our system using Python 3.7 (Anaconda 2019.10 distribution) under Linux (Xubuntu 18.04 LTS). For the pix2pix implementation we used Keras library with Tensorflow backend.

The input melody format is in the MIDI format which is a protocol for the communications of data between devices. A MIDI file has instructions about how the song is played and information about the notes of the musical piece.

The dataset we used is the “classical piano MIDIs” dataset from [11], from which, we chose the music of Chopin to perform our training; they total to 48 piano MIDIs from the famous Chopin’s musical pieces. For each musical piece, we created a corresponding modified version that has only the main melody removing the accompaniment music. We divide each midi file into a number of chunks, each of which is later converted into an image with the help of “midi2img” [12]. We divided each MIDI file into chunks of length sixteen seconds each. Converting the MIDI chunks into images resulted in a total of 398 images; we reserved 340 of them for training and 58 for testing. Each image containing a main melody has a corresponding (one-to-one) image that has the color encoded accompaniment music.

Our experimental results show that our suggested data representation method through color encoding the accompaniment music achieved better results in the pix2pix GANs than the b/w representation. Fig. 5 shows a comparison between b/w representation for the accompaniment music and our color encoding representation. Fig 5(a) represents the loss curve of the total loss represented by Eq. (1) during training the pix2pix with b/w images of the accompaniment music. Fig. 5(b) shows the total loss curve during training with our color encoded images. Fig. 5(c) and (d) show the loss curves of the generator loss as in Eq. (2) during training the pix2pix with b/w and color encoded images respectively. To plot these graphs, we recorded the loss values at predefined checkpoints.

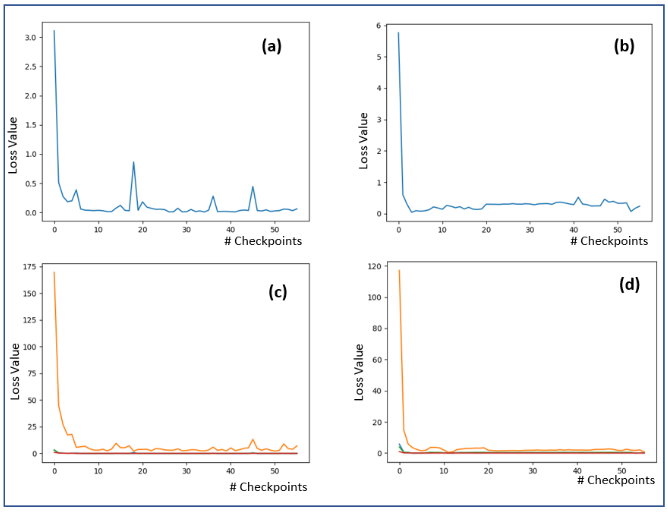

Fig. 5. The loss curves for pix2pix GANs trained with b/w images (left column) versus our coloring schemed images (right column)

The plotted graphs show that, in case of training the pix2pix network with b/w image representation of the accompaniment music, the loss curves are unstable during training in opposite to the loss curves during the training with our color encoded images. It is also recognizable that the loss values at the end of the training are lower for the encoded images training (0.001) than those for the b/w training (0.02).

Fig. 6. shows a sample output from the accompaniment music generation pix2pix GAN after training with b/w images versus our color encoded images after only 8 epochs. It is clear that the former failed to learn as opposed to the latter which converged in few epochs.

V. Discussion

Pix2pix GANs can be used to transfer style between images having similar structure. This transfer is ensured by the cross-entropy loss function of the generator described in Eq. (2). However, representing the accompaniment music in the traditional way, adopted in the literature [7, 8, 9], produces images that have a totally different structure. Consequently, the network fails to learn the correlation between the input and target images in most of the cases. In our case, the network failed to learn the correlation between the input main melody images and the target accompaniment music images.

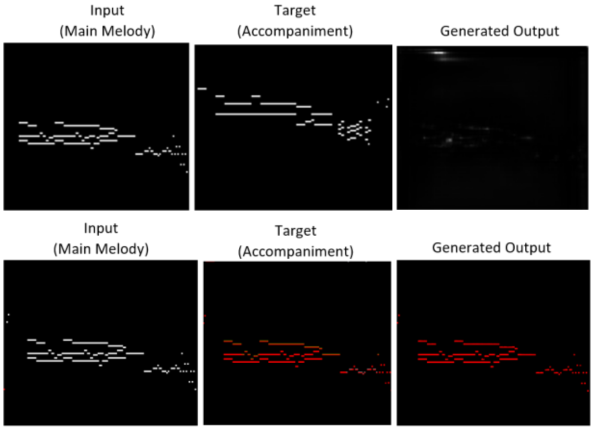

Fig. 6. Samples from the output of the pix2pix training with b/w representation (upper row) versus our color encoded representation (lower row).

On the contrary, our proposed color encoding representation ensures that the structure of the input image is reserved in the target image while we still store the accompaniment music position information in the form of colors, this way the network succeeded in the learning process.

The accompaniment music generated from our system is promising although it has some notes that are not necessarily in harmony with the input main melody. As an enhancement, we need to do post-processing to ensure the harmony between input and generated music. Another suggestion is that maybe we can store some kind of harmonic information inside the image representation of music.

VI. Conclusion and Future Work

In this paper we proposed a system for intelligent generation of accompaniment music using pix2pix GANs. We started by giving a brief theoretical background on the pix2pix GANs highlighting the recent applications in the field of computer music generation. We described our methodology focusing on our novel data representation technique for encoding accompaniment music as colors inside the main melody images. We exhibited our experimental results comparing them to the results of training the pix2pix with the standard b/w representation of the accompaniment music (binary piano-rolls adopted in earlier literature). The results showed that pix2pix GANs have difficulties in learning the traditional b/w representation. On the contrary, using our proposed data representation technique, the network started to converge in few epochs with a generator loss value as low as 0.001. The limitation to our work is that the generated accompaniment music still needs some enhancement to make it more harmonic with the input main melody.

There are various future work opportunities that can make use of our work; utilizing pix2pix GANs in music opens the door for many application ideas, not only for generating accompaniment music but also for generating bassline, rhythm or even for music orchestration. Moreover, our proposed music representation can be applied in training other types of deep neural networks such as cycleGANs or even the traditional GANs for producing music that resembles famous composers or even combine styles of different composers together. In the future we will study enhancing the harmony between the generated accompaniment music and the input melody through post-processing or through storing harmony information in the image representation of music.

References

[1] Jin, Cong & Tie, Yun & Bai, Yong & Lv, Xin & Liu, Shouxun. (2020). A Style-Specific Music Composition Neural Network. Neural Processing Letters. 10.1007/s11063-020-10241-8.

[2] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. “MIDINET: A convolutional generative adversarial network for symbolic-domain music generation”, In International Society of Music Information Retrieval(ISMIR), Suzhou, China, 2017.

[3] María Navarro-Cáceres, Marcelo Caetano, Gilberto Bernardes, Leandro Nunes de Castro, and Juan Manuel Corchado. “Automatic generation of chord progressions with an artificial immune system”, In Colin Johnson, Adrian Carballal, and João Correia, editors, Evolutionary and Biologically Inspired Music, Sound, Art and Design, pages 175-186, Cham, 2015. Springer International Publishing.

[4] Kanae Kunimatsu, Yu Ishikawa, Masami Takata, and Kazuki Joe. “A music composition model with genetic programming -a case study of chord progression and bassline-“, In International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA'15, pages 256-262, Monte Carlo Resort, Las Vegas, USA, July 27-30, 2015.

[5] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1125–1134). 49, 98, 100, and 103

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2014). Generative Adversarial Nets. In Advances in Neural Information Processing Systems (NIPS) (pp. 2672–2680). xi, 12, 45, 96, 97, 100, and 102.

[7] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In AAAI Conference on Artificial Intelligence, Louisiana, USA, 2018.

[8] H. Dong and Y. Yang. Convolutional generative adversarial networks with binary neurons for polyphonic music generation. CoRR, abs/1804.09399, 2018.

[9] M. Oza, H. Vaghela, and K. Srivastava. Progressive generative adversarial binary networks for music generation. ArXiv, arXiv:1903.04722, 2019.

[10] Jong Wook Kim and Juan Pablo Bello. Adversarial learning for improved onsets and frames music transcription. In International Society for Music Information Retrieval Conference, pages 670–677, 2019.

[11] Classical Piano Dataset: https://www.kaggle.com/soumikrakshit/classical-music-midi?select=chopin. Last accessed: July 29th, 2020.

[12] img2midi: https://github.com/mathigatti/midi2img. Last Accessed: July 29th, 2020.