Generate Image from Music Emotion

Zhang Xian

SID: 1155193168

Project report for the course

AIST4010 Foundation of Applied Deep Learning

ABSTRACT

In this project, I applied my knowledge of deep learning and music information retrieval to support generating images from music emotion. Generative Adversarial Networks (GAN) and Diffusion Model are both explored, discovering that the latter yields superior results. The pipeline implements the pretrained Random Forest model and the Diffusion model that predicts the potentially valence and arousal emotion values of music and generates an image according to the predicted emotion values, respectively. Python is the major programming language for model training and data management. The Valence-Arousal Model is the connection between music information retrieval and image generation to convey the music emotion to the image emotion.

1. Background

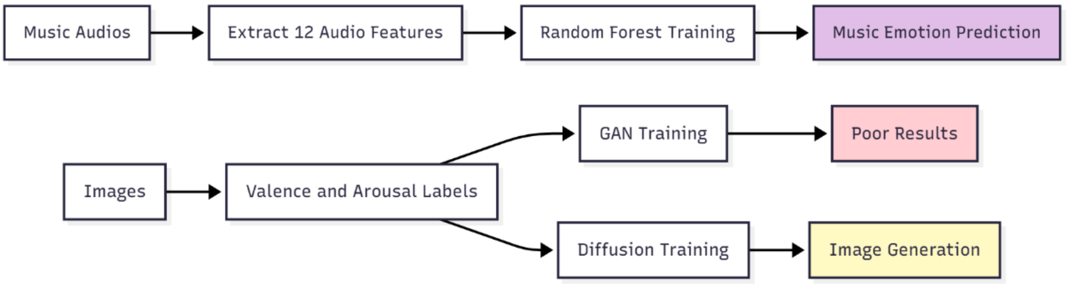
Cross-modal artificial intelligence can create meaningful connections between different forms of data. This project applies my knowledge by designing a deep learning pipeline that translates the emotional content of music into a visual image representation. The process bridges two distinct domains, which are music audio signal processing and image generation.

The Arousal-Valence Model is the connection between music information retrieval and image generation, conveying music emotions to image emotions. It is a continuous two-dimensional framework where arousal indicates intensity and valence represents sentiment. The pipeline implements a Random Forest model to predict these valence and arousal emotion values from music, which is a regression task.

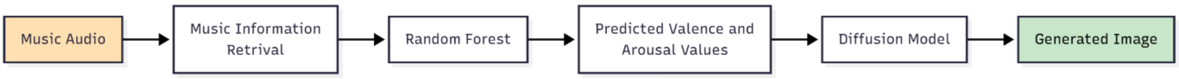
A Generative Adversarial Network (GAN) and a Diffusion Model were both explored for generating an image according to the predicted emotion values, discovering that the Diffusion Model has superior results. This stage completes the pipeline, which uses the Diffusion Model to generate an image from the emotion values derived from the input music by the Random Forest.

2. Methodologies

The following flowcharts demonstrate the progress of training.



The following flowchart shows the overall pipeline of the image generation from input music.


Music Audio
Music Information Retrival
Random Forest
Predicted Valence and Arousal Values
Diffusion Model
Generated Image

2.1. Programming Languages

I use Python for both data cleaning and model training. The following Python libraries are included in the project:

- Pandas
- NumPy
- Librosa
- Scikit-learn
- Optuna
- PyTorch
- Joblib
- Pillow (also as Python Imaging Library)
- Matplotlib

2.2. Data Source and Cleaning

Data is significant in model training. Music data and corresponding valence and arousal values are required for training the random forest model for predicting music emotion. Image data is required for training the GAN model. The raw data sources are listed below:

Music Data

- DEAM dataset - The MediaEval Database for Emotional Analysis of Music [1]
- PMEmo: A Dataset for Music Emotion Computing [2]

Image Data

- Image-Emotion (Arousal and Valence) CGnA10766 Dataset [3]

All the mentioned data sources provide manually labelled valence and arousal values for each data item, which is crucial for music and image emotion representation. Data cleaning and manipulation skills are adapted to filter out the invalid data and combine datasets into one. For the image dataset, a subset of 1311 landscape images was manually selected from the original dataset by me. This selective filtering was performed to narrow the domain scope, thereby reducing the complexity of the feature space the generative model needs to learn. By focusing exclusively on landscape images, I aimed to speed up the model convergence and improve generation fidelity within the constraints of limited computational resources.

2.3. Music Information Retrieval

To analyse the music emotion information, a comprehensive view of the music is required. Librosa contains several functions to get specific information from the music. The retrieved music information is cleaned and arranged into a pandas DataFrame for feeding into the random forest model. The overall music information retrieval takes 1 hour and 44 minutes. In the project, I used librosa to extract the following features.

- Mel-Frequency Cepstral Coefficients
- Chroma Features
- Spectral Contrast
- Zero-Crossing Rate
- Tempo
- Root Mean Square Energy
- Spectral Centroid
- Spectral Bandwidth
- Spectral Rolloff
- Tonal Centroid Features
- Chroma Constant-Q Transform
- Chroma Chroma Energy Normalised

2.4. Random Forest

The random forest is a machine learning model which combines many decision trees to predict values. Each decision tree makes a prediction. The final decision is the average of all these predictions. An equation to represent the random forest is

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

where B is the number of trees and $f_b(x')$ is the prediction made by tree b. The random forest model is trained using the retrieved music information to predict the valence and arousal values.

2.5. Generative Adversarial Network

Generative adversarial networks are composed of two models, the generator and the discriminator. The generator generates results while the discriminator checks the results generated by the generator to detect the accuracy of the results. In the project, both the generator and the discriminator are a convolutional neural network (CNN). CNN can generate an image layer by layer and decompose it layer by layer. The CNN-composed GAN can generate images more accurately with the checks of the discriminator on the generator.

Emotional Residual Unit

Merely having a GAN model is not enough for generating an emotional image. The GAN must generate images based on the intake valence and arousal values. To achieve this, an emotional residual unit (ERU) was proposed by Park *et al.* [4] to enhance the matching of the emotion performance of the generated image with the valence and arousal values. The mechanism first concatenates the valence matrix V filled with all the same valence value and the arousal matrix A with all the same arousal value as new channels to the input map generated M, respectively, to form maps [V, M] and [A, M], where [] means concatenation. Then, the two maps are formed by passing through convolution layers to match the same dimension as the input map M. After this, the maps must pass through the sigmoid activation function $Sigmoid(x) = \frac{1}{1 + e^{-x}}$. So far, there are two new maps, let them be v and a.

$$v = Sigmoid\ (\ Convolution\ [Valence\_Map\_V, Generated\_Image]\ )$$

$$a = Sigmoid\ (\ Convolution\ [Arousal\_Map\_A, Generated\_Image]\ )$$

Next, the maps have to element-wisely multiply M and add together to form map B.

$$B = (v \otimes M) \oplus (a \otimes M)$$

Lastly, N concatenates the concatenation of valence and arousal matrices V and A. Their concatenation passes through another convolution layer, followed by the tanh activation function $Tanh(x) = \frac{1- e^{-x}}{1 + e^{-x}}$ to output the output map E of ERU.

$$E = Tanh\ (\ Convolution\ \big[B, [Valence\_Map\_V, Arousal\_Map\_A]\big])$$

Affective Feature Matching Loss

To calculate the loss of the GAN model, an affective feature matching loss is introduced by Park *et al.* as a method to optimise the model. The mechanism is simply to compare the difference between the feature map of the generated image and the real image. Remember the maps v and a in the ERU unit. They are the feature maps of the generated image in valence and arousal values, respectively. The discriminator has to find the same feature maps of the real image with the same valence and arousal values as the generated image. The affective feature matching loss is the difference between the two sets of feature maps.

$$Loss_v = \frac{1}{k}\sum_{i=1}^{k} |v - Sigmoid\ (\ Convolution\ [Valence\_Map\_V, Real\_Image]\ )$$

$$Loss_a = \frac{1}{k}\sum_{i=1}^{k} |a - Sigmoid\ (\ Convolution\ [Arousal\_Map\_A, Real\_Image]\ )$$

2.6. Diffusion Model

Apart from the GAN, a Denoising Diffusion Probabilistic Model (DDPM) is also explored for image generation. The core architecture is an Emotion-Conditioned U-Net enhanced with residual connections and self-attention mechanisms. Unlike GAN, the diffusion model's objective is not to generate an image directly, but to iteratively predict and remove noise.

During the training, the raw images are progressively corrupted by adding Gaussian noise across $T = 1000$ timesteps, transitioning from a clean state to pure noise. The noisy images x_t are fed into the U-Net. Simultaneously, the continuous valence and arousal labels are projected into a 256-dimensional emotion embedding vector E_{emb} via a Multilayer Perceptron.

Emotion Conditioning (AdaIN)

An Adaptive Instance Normalization (AdaIN) mechanism is introduced to integrate emotion into the visual generation. At each convolution block of the U-Net, the emotion embedding is projected to learn a channel-wise scale (γ) and shift (β) parameters. These parameters modulate the intermediate feature maps H, effectively steering the generation process towards the desired emotional attributes

$$H_{new} = H_{old} \times \left(1 + \gamma(E_{emb})\right) + \beta(E_{emb})$$

Classifier-Free Guidance (CFG)

A Classifier-Free Guidance strategy is introduced to strengthen the alignment of the image and input valence and arousal emotions. The model performs two predictions at each step, which are a conditional prediction ϵ_{cond} with VA and an unconditional prediction ϵ_{uncond} without VA. The final noise estimate is derived by amplifying the difference between them.

$$\hat{\varepsilon} = \epsilon_{uncond} + s \times (\epsilon_{cond} - \epsilon_{uncond})$$

Where s is the guidance scale, which is a hyperparameter. This forces the model to prioritize features that are uniquely associated with the specified emotion.

Objective Function

The model is optimized using the Mean Squared Error (MSE) loss, which measures the discrepancy between the actual added noise ϵ and the predicted noise ϵ_θ.

$$\mathcal{L} = \left|\left|\varepsilon - \varepsilon_\theta(x_t, t, v, a)\right|\right|^2$$

3. Performance

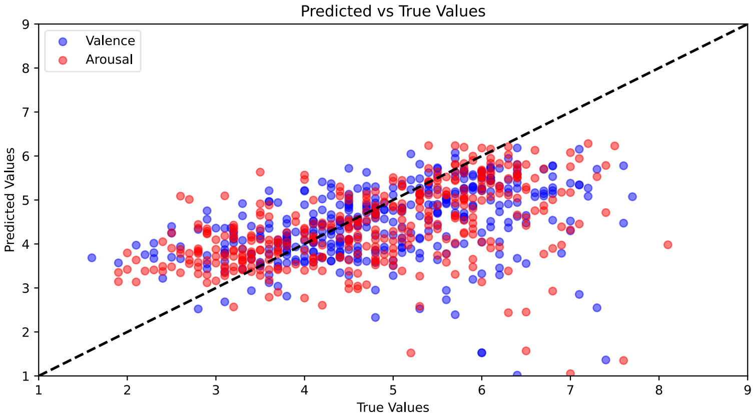
Random Forest

Optuna [5] is an automatic hyperparameter optimisation software framework which is solid in fine-tuning models by finding optimal hyperparameters. In the project, Optuna is used for training the random forest to achieve the optimal performance. The optimal hyperparameters provided by optuna are in the table below.

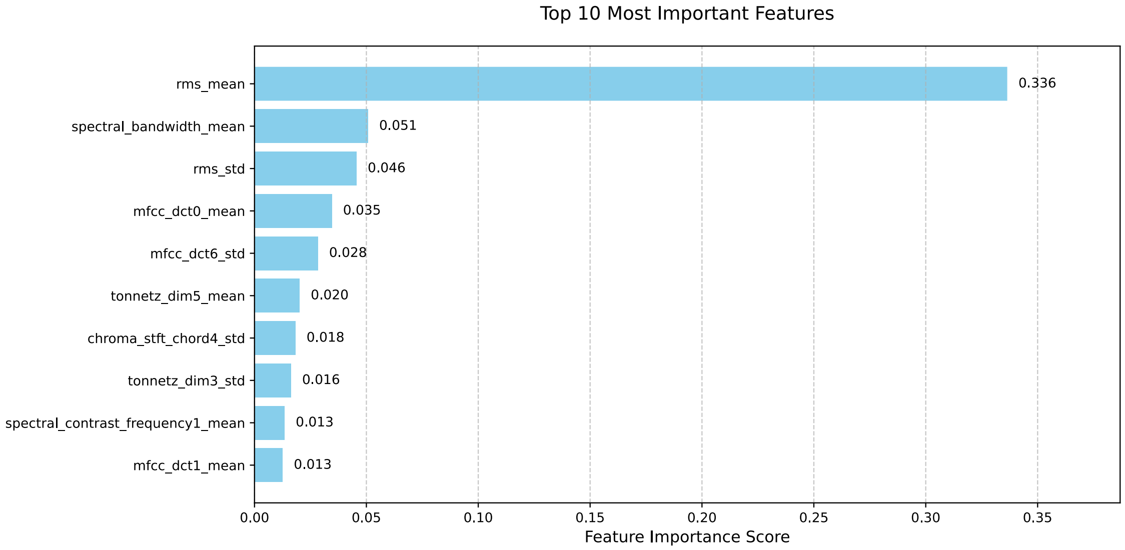
Number of trees	Max Depth	Minimum number of samples to split a node	Minimum number of samples of leaf node	Max feature numbers
600	20	2	3	1

The table below shows optimal evaluation results.

Mean Squared Error	Mean Absolute Error	Root Mean Squared Error	R^2 Score	Explained Variance Score
1.81242	0.96088	1.34626	0.60855	0.60878



The above graph shows the predicted and real valence and arousal values. From the true values axis, which is the x-axis. The range of the data varies from approximately 1.5 to 8. Which is acceptable. There is a concern about biased data due to the lack of extreme emotions. Data augmentation is a skill to increase the data range for better generalisation of data. However, it is not promoted in the project because emotion is already subjective to humans and varies a lot between individuals. Augmenting the data range may not reflect the true emotion of a person, thus, it loses the authenticity and facticity.

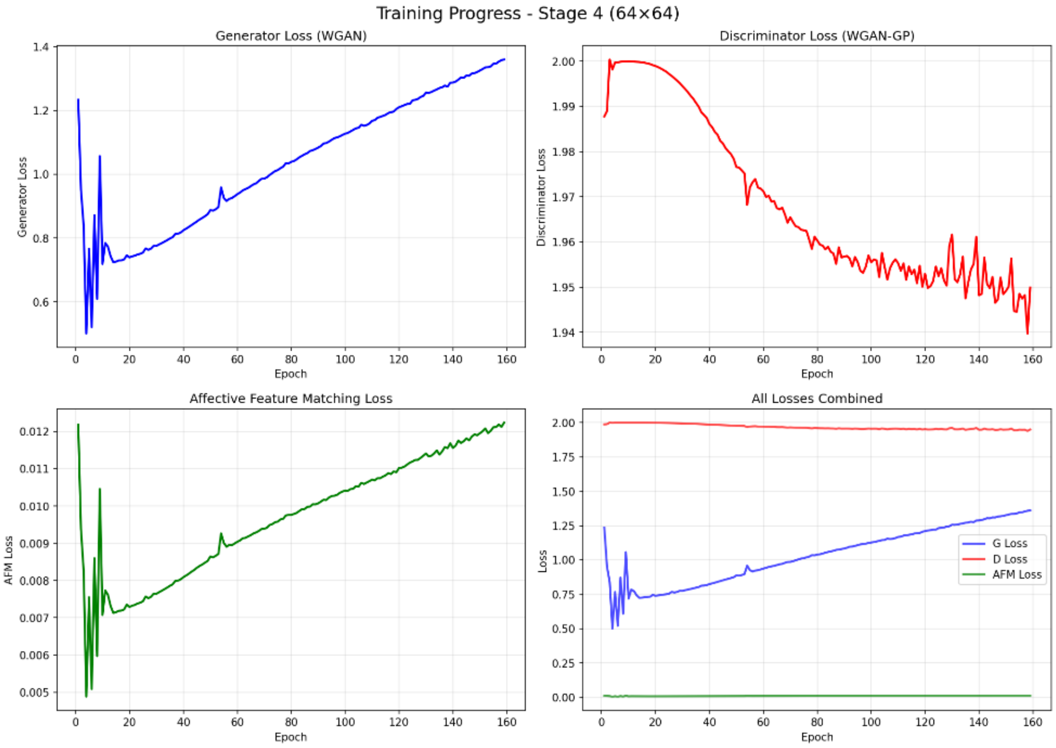


The above graph shows the top 10 most important features according to the weights in the training of the random forest model. The most important feature is the root mean square energy of music. It can suggest that the loudness of the dynamics of the music has a significant role in determining the emotions of the music. Which makes some sense because the dynamics between happy and sad music have huge differences.

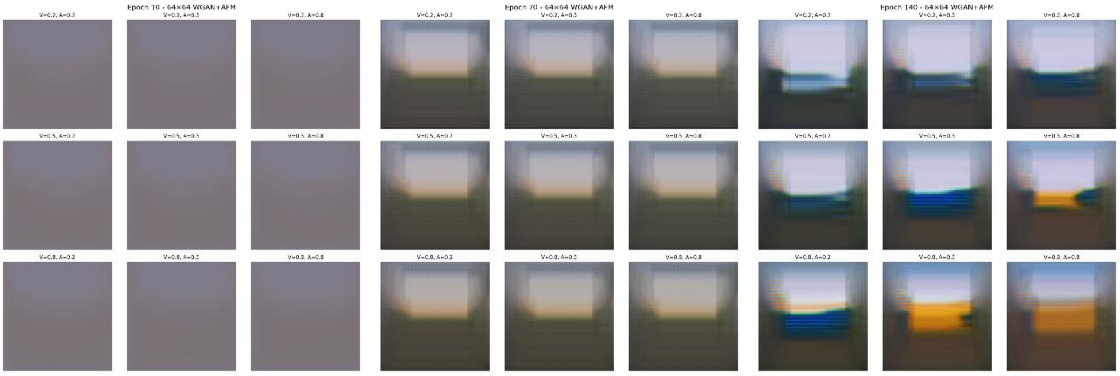
Generative Adversarial Network

I first tried using a WGAN-GP model to generate images from the valence and arousal values. The introduced Emotional Residual Unit and the Affective Feature Matching Loss are implemented in the training. The goal was to see if the GAN could learn to map emotions to landscape features. However, the results were not good. Even with the gradient penalty to help stabilize training, the model had a serious problem with mode collapse. The model kept producing very similar and repetitive images, no matter what the emotion input is. The discriminator learned too fast, which made it hard for the generator to improve.

To try and fix these problems, some advanced techniques are used to improve the model, such as a progressive growing strategy. The model is trained from very small images of 4×4 pixels and slowly increased the size up to 128×128 pixels to make the training more stable. Even with these advanced methods, the training was still very unstable.



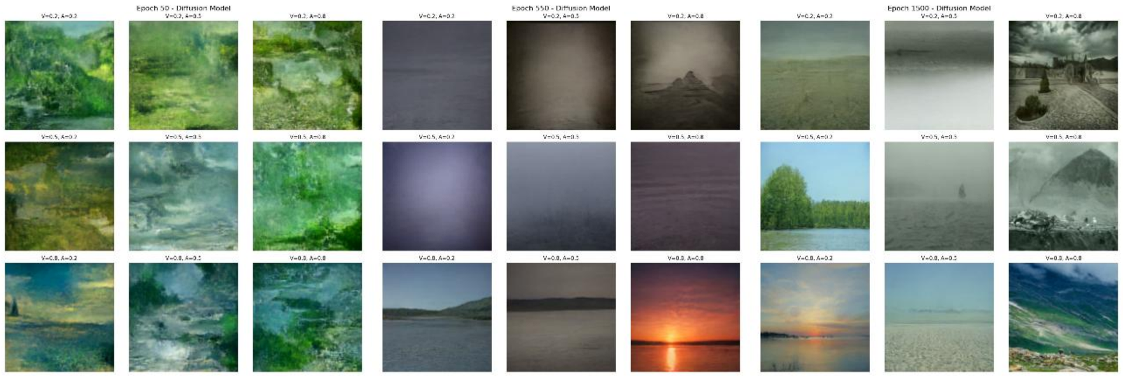
The above figure is the training loss curves. The red line (discriminator) drops to zero, while the blue line (generator) goes up, showing that the model stopped learning. The training graphs reveal a major problem. The discriminator's loss dropped very fast, while the generator's loss kept going up. This means the discriminator became too smart too quickly. It easily spotted the fake images, so the generator stopped receiving useful feedback on how to improve.



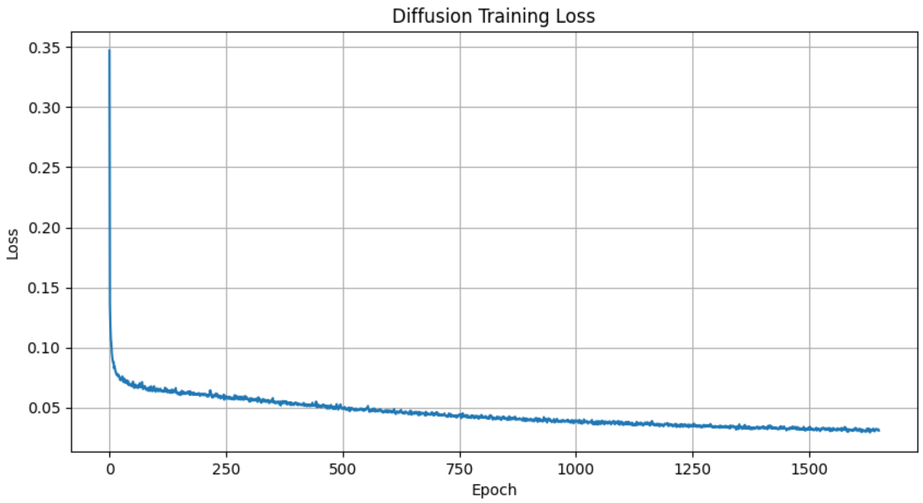
As a result, the Generator failed. As seen in the above figure, it started producing the same blurry, brownish images over and over again, no matter what valence and arousal values are used. This is a common mode collapse problem. After so many experiments, because the GAN could not create diverse images or follow the emotion rules, I decided to switch to Diffusion Models. Diffusion models are much more stable and are better at creating high-quality, diverse images that actually match the user's feelings.

Diffusion Model

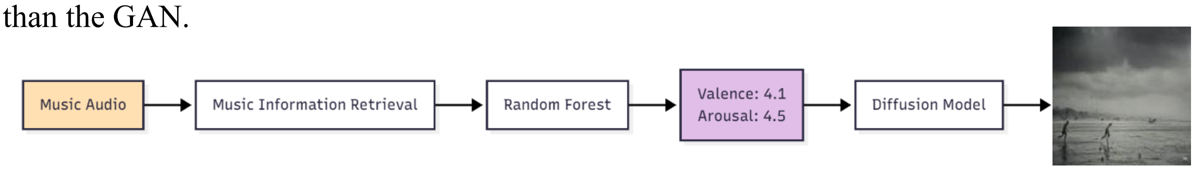
After the GAN model failed to produce good results, I switched to the Denoising Diffusion Probabilistic Model. The model is trained for more than 1500 epochs and observed a clear, steady improvement in quality that was completely different from the chaotic training of the GAN. At the beginning, around Epoch 50, the model was just learning basic colors and shapes, producing mostly blurry blobs of green and blue. By Epoch 550, it had started to understand structure, creating visible horizon lines and separating the sky from the ground, although the images remained somewhat dark and hazy as it was still refining how to handle lighting.



By the end of training at Epoch 1500, the results were a lot better and showed that the model had learned to map human emotions to visual landscapes somehow. The images became sharp and detailed, more accurately reflecting the specific valence and arousal values. For example, when I input high valence and low arousal, the model generated a calm, peaceful sunset over water, whereas high arousal inputs produced energetic, bright green mountain scenes, and low valence inputs resulted in dark, stormy landscapes. This diversity proved that the model was not just memorizing data but actually understanding the emotional context.



The above figure is the training loss of the model. Unlike the GAN experiment where the loss fluctuated wildly, the Diffusion model's loss dropped quickly at the start and continued to decrease steadily throughout the entire process. This stability meant the training was reliable and stress-free. Comparing the two experiments, the diffusion model is clearly a lot better than the GAN.



The above flowchart shows the demo of the image generation from music by using the trained random forest and diffusion model.

4. Challenges

One of the most persistent challenges observed is that the diffusion model has a strong bias towards desaturated, grey-toned images. Vivid, colorful landscapes are only generated when the input valence and arousal values are extremely high, while middle-range values consistently result in uniformly gloomy outputs. This suggests the model learned to associate neutral emotions with a lack of color rather than just different lighting or scenery. Additionally, the generated images often lacked diversity, frequently producing similar compositions even when different random seeds are applied. This issue is likely driven by the low variance in the landscape training data. Without diverse examples of specific emotional scenes, the model struggles to generalize effectively.

Theoretical limitations regarding the subjectivity of music emotion also posed a significant hurdle. It is difficult to determine the optimal number of music features to capture complex feelings, and reducing human emotion to simple valence and arousal values can be monotonous and oversimplified. Since music perception is highly subjective, what one listener finds sad, another might think of as nostalgic. This subjectivity introduces ambiguity in the project's two-stage mapping approach. If the initial music-to-emotion prediction is inaccurate, that error propagates to the image generation stage, creating a disconnect between the audio and the visual result.

Finally, the high computational cost of training proved to be a major bottleneck. The diffusion model required 1500 epochs and even more to achieve high-quality results, consuming substantial GPU resources and time, not to mention the time cost of unstable retraining. This high cost severely limited the scope for extensive hyperparameter tuning or testing different architectures.

5. Conclusion

To conclude, I tried to link the connection between music and image through emotions. I am also trying to build a website prototype for easier use. I experienced a lot of joy in the project and strengthened my understanding of deep learning and cross-modal generation. Even though the final result so far of the image generation is not very accurate, I will still try to improve it and wish that one day I can show a decent result.

6. Reference

[1] “DEAM dataset - Database for Emotional Analysis of Music,” cvml.unige.ch. https://cvml.unige.ch/databases/DEAM/

[2] K. Zhang, H. Zhang, “GitHub - HuiZhangDB/PMEmo: PMEmo: A Dataset For Music Emotion Computing,” GitHub, 2018. https://github.com/HuiZhangDB/PMEmo.

[3] Y.-S. Kim, “CGnA10766 Dataset,” figshare, Jun. 2018, doi: https://doi.org/10.6084/u002Fm9.figshare.5383105.v1.

[4] C. Park and I.-K. Lee, “Emotional Landscape Image Generation Using Generative Adversarial Networks,” Lecture notes in computer science, pp. 573–590, Jan. 2021, doi: https://doi.org/10.1007/978-3-030-69538-5_35.

[5] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” in Proceedings of the KDD Conference, 2019.