**R.V. COLLEGE OF ENGINEERING**

**BENGALURU – 560059**

(Autonomous Institution Affiliated to VTU, Belagavi)



**“COMPARATIVE ANALYSIS OF MUSIC STYLE TRANSFER”**

**DSML (16CS72) LAB PART B REPORT**

**Submitted by**

**Name USN**

**SHREYA SAHAY 1RV17CS200**

**JAINIL VIREN PARIKH 1RV17CS063**

**AMISHA 1RV17CS017**

**KARTHIK KS 1RV17CS071**

**ABHISHEK R 1RV17CS005**

**Under the Guidance of**

**Mr. Karthik K Natrajan**

**NOKIA**

**Department of Computer Science & Engineering**

**ABSTRACT**

Observing the success of neural nets in other fields this project aims to produce interesting results from music . Music is fundamentally a sequence of notes. A composer constructs long sequences of notes which are then performed through an instrument to produce music. Humans can easily identify the genre of a music by just listening to it .However it is quite difficult to parametrize the style of music as it is not dependent on any fixed metric such as pitch,vocal etc but depends on its composition and performance which need to be correlated. Although the genre of music (Classical , Jazz etc )depends on the listener, there are certain distinguishing characteristics of each genre .Generative models can be applied to change properties of existing data in a principled way, even transfer properties between data samples.In this report we will aim to understand the styles of the different music genres and transform one style of music to another through transfer of properties. This is a comparison between two approaches and the resulting musical pieces, namely, Autoencoder-Generative Adversarial Networks (VAE-GAN) and Spectrogram Analysis methods .

**Table of Contents**

**Abstract i**

**Table of Contents ii**

**List of Figures iii**

[**Chapter 1**](#_heading=h.3znysh7) **[1](#_heading=h.3znysh7)**

**[Introduction](#_heading=h.3znysh7) [1](#_heading=h.3znysh7)**

[1.1 Motivation](#_heading=h.2et92p0) [1](#_heading=h.2et92p0)

[1.2 Problem Statement](#_heading=h.2et92p0) [1](#_heading=h.2et92p0)

[1.3 Literature Review](#_heading=h.2et92p0) [1](#_heading=h.2et92p0)

[1.4 Methodology](#_heading=h.tyjcwt) [2](#_heading=h.tyjcwt)

[Organization of Report](#_heading=h.3dy6vkm) [3](#_heading=h.3dy6vkm)

[**Chapter 2**](#_heading=h.1t3h5sf) **[5](#_heading=h.1t3h5sf)**

**[High Level Design of System](#_heading=h.1t3h5sf) [5](#_heading=h.1t3h5sf)**

[2.1 System Compositions](#_heading=h.4d34og8) [5](#_heading=h.4d34og8)

[2.2 Input Design](#_heading=h.4d34og8) [5](#_heading=h.4d34og8)

[2.3 Software Design](#_heading=h.4d34og8) [5](#_heading=h.4d34og8)

[2.3.1 High Level Software Design](#_heading=h.17dp8vu) [5](#_heading=h.17dp8vu)

[2.3.2 Low Level Software Design](#_heading=h.17dp8vu) [6](#_heading=h.17dp8vu)

[**Chapter 3**](#_heading=h.3rdcrjn) **[7](#_heading=h.3rdcrjn)**

**[Implementation](#_heading=h.3rdcrjn) [7](#_heading=h.3rdcrjn)**

[3.1: Input Implementation Software](#_heading=h.26in1rg) [7](#_heading=h.26in1rg)

[3.1.1: Data Collection](#_heading=h.lnxbz9) [7](#_heading=h.lnxbz9)

[3.1.2: Input Software Algorithm](#_heading=h.lnxbz9) [7](#_heading=h.lnxbz9)

[3.1.2.1: Spectrogram Preprocessing :](#_heading=h.35nkun2) [7](#_heading=h.35nkun2)

[3.2 Implementation of Individual Components](#_heading=h.1ksv4uv) [7](#_heading=h.1ksv4uv)

[3.2.1 GAN architecture:](#_heading=h.44sinio) [7](#_heading=h.44sinio)

[3.2.2 VAE-GAN Training :](#_heading=h.2jxsxqh) [8](#_heading=h.2jxsxqh)

[3.2.3 Spectrogram analysis:](#_heading=h.z337ya)

3.2.4 Web Interface Implementation

[**Chapter 4:**](#_heading=h.3as4poj) **15**

[4.1 : Results and observations](#_heading=h.4i7ojhp) 15

[4.1.1 Gan Results:](#_heading=h.2xcytpi) 15

[**Chapter 5**](#_heading=h.3whwml4) **[12](#_heading=h.3whwml4)**

**[5.1 CONCLUSION](#_heading=h.3whwml4) 19**

[5.2 Future Work](#_heading=h.2bn6wsx) 19

[**References**](#_heading=h.qsh70q) **20**

**List of Figures**

**Figure No. Figure Name Page.No.**

2.1 System Compositions 9

2.2 Input Design 9

2.3.1 (a) High Level Design for 10

Spectrogram

2.3.2 (b) High Level Design for 10

VAE-GAN

2.3.2 (a) Low Level Design for 11

Spectrogram

2.3.2 (b) Low Level Design for 11

VAE-GAN

4.1.1 (a) VAE-GAN training 15

4.1.1 (b). VAE-GAN losses while training. 15

4.1.1 (c) VAE- GAN results after 16

thresholding

4.1.2 (a) Input for Spectrogram 17

4.1.2 (b) Output for Spectrogram 17

4.1.3 (a) Input Web Page 18

4.1.3 (b) Output Web Page 18

# **Chapter 1**

# **Introduction**

## 1.1 Motivation

The procedure of crafting music or algorithmic composition has become popular these days due to the sudden increase in CPU powers which helped researchers to try newer approaches quickly. Music style Transfer which is a part of algorithmic composition, focuses specifically on changing the characteristics of a song such that it sounds as if it were sung by another musician.Companies like Facebook have successfully been able to classify music i.e to differentiate music of one genre from the other ,we aim to recreate music of a specific genre by understanding the characteristics of different genres. Recent breakthroughs in deep generative models has led to the use of GAN’s in automotive music generation. In this project we explore two different techniques both employ deep neural networks to understand the intricacies involved in musical notes. Despite these promising methods it is to be noted that both natural and creative music generation is not feasible as an algorithm with weak constraints is often too random and rarely generates human-like music and an algorithm with strong constraints is too flat and they lack the dynamic that can be felt in naturally created music.

## 1.2 Problem Statement

The problem is to develop a system that can learn the features that define the genre of music and manipulate the parameters extracted from the music so as to bring about a change in the genre through two methodologies- a) Spectrogram Analysis and b) VAE-GAN method and output the two resultant music files to the user through a suitable interface and also to compare the performances of the two methods.

## 1.3 Literature Review

The idea of interpreting and building an understanding of the artistic styles of various pieces of art has been an area of interest since the beginning of 2010s. This has especially been explored and developed in the image domain.

The groundbreaking work in this aspect[1] looks at decomposing an image into its style and content and recombining them to produce a result. This decomposition and recombination paved the way to understanding how deep neural networks can be made to learn artistic features and obtain their neural representations in spatial domain. Higher layers of a deep convolutional neural network that is trained for object recognition essentially capture the content information and the texture feature space contributes to the style information. Following this, an attempt was made at providing a style migration model[2] that improved on the per pixel losses as used in [1].

Once the use of CNNs in neural art style transfer reached a sense of maturity in terms of the research carried out, its potential applications caught the eyes of music scientists who were curious to see whether and how the highly capable neural networks can understand the artistic styles in music in addition to the sequence of notes and pitches.

The very first stage in this direction was the birth of music genre classification networks that could derive some latent representation for the style of music which is not mathematically parameterizable. The only way to characterize a music genre is by the common features shared by its members. The paper[3] discusses the classification of music into genre hierarchies utilizing both whole-frame as well as real time scenarios using beat histograms to represent the styles. This paper[4] uses harmonic structures like chords to classify Brazillian music. It establishes a random forest model for classification into nine genres. Similarly, multiple approaches that optimize the genre classification have been proposed and are seen in many commonly used applications like spotify, soundcloud, shazam.

With a proper genre classification system in place, research interest progressed towards generating music and further control the style of music being generated.

The paper[5] proposes a scientifically-viable definition of music style transfer by breaking it down into precise concepts of timbre style transfer, performance style transfer and composition style transfer. This paper[6] proposes a variational autoencoder setup to handle music with multiple music tracks to change the pitch, dynamics and instruments in order to convert an input music from one style to another, [7] developed a system based on a WaveNet autoencoder [8] that can translate music across instruments, genres and styles, and even create music from whistling. The use of autoencoders, as in [9] based on a recurrent highway gated network combined with a variational autoencoder with filtering heuristics allows generating pseudo-live acoustically pleasing and melodically diverse music.

## 1.4 Methodology

The proposed style transfer is done in two methods:

1. VAE- GAN method- This method advocates the combined use of variational auto encoder and generative adversarial network. The encoder is composed of conv2D, batch normalization and dropout layers that take as input a batch of images of size 88x88 and produces an intermediate representation which acts as an input to the decoder. The decoder uses a series of conv2D transpose layers. The learned feature representation in GAN discriminator is used as a basis for construction of the output music file by the VAE.
2. Spectrogram Analysis- The given input music files are converted into spectrograms.To extract features from the spectrogram images we require a neural network that can understand the complex patterns of the spectrogram image. A neural network with random weight initialisations was used for the same. Then using L-BFGS we update the weights(random noise vector) such that the loss is minimized. The resultant spectrogram is initialised to random noise and later processed to combine the feature information of the spectrograms of the input music. The resultant spectrogram is then converted to a music file delivered as output.

A web interface is also implemented to facilitate end to end user interaction with the two models. The interface takes two audio files as input and provides them as input to the models. After the models synthesize the output files, both the output files are played in a web page with appropriate audio controls.

# Chapter 2

# High Level Design of System

## 2.1 System Compositions

## Fig. 2.1

## 2.2 Input Design

## 

## 

## Fig. 2.2

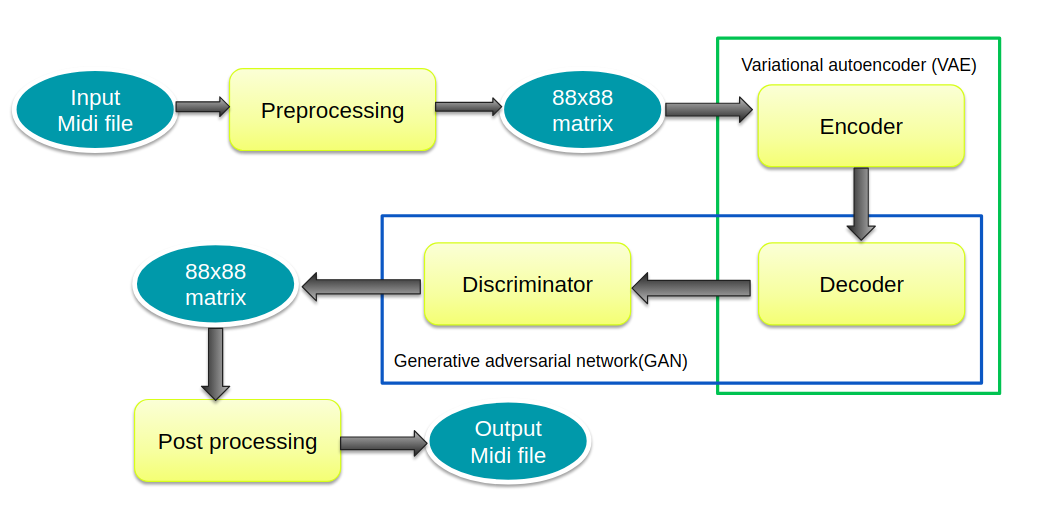
## 

## 2.3 Software Design

### 2.3.1 High Level Software Design

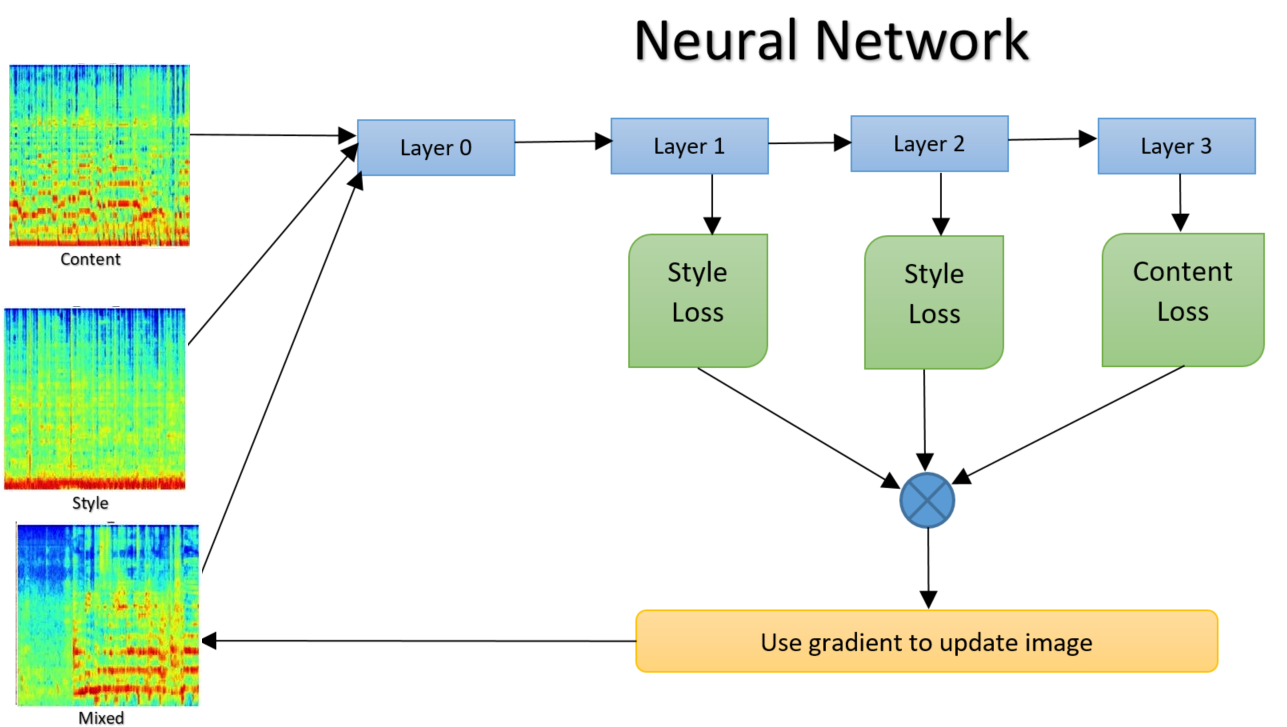
### https://lh6.googleusercontent.com/6b4Dq7s-nw_lZ3K9ru1Tg58ud2TkQbh_diMJc9NGlJTA8vN0-cDo-qOp3iBxCw0pKqXlpocQwgrXZlf_3_lVHMqdGJj_Cb2Zy0SsEA0WIN95UXx0gOB5edfzD-1zBFjBsZDPyCiG

**Fig. 2.3.1(a)**

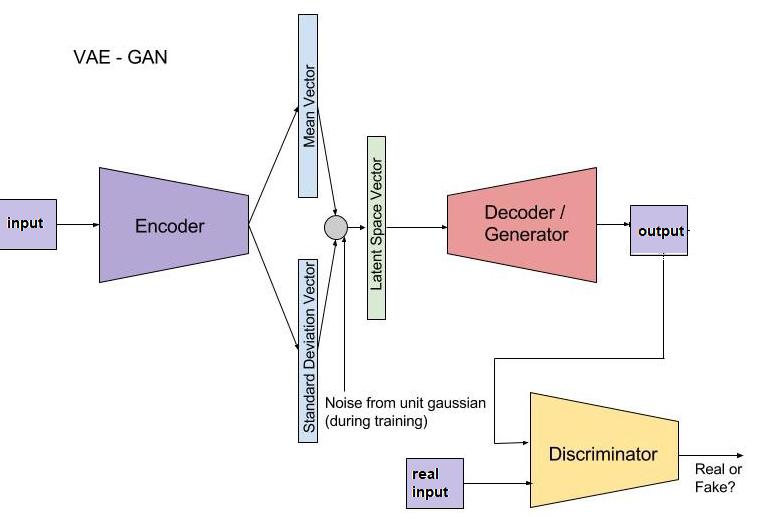


**Fig. 2.3.1(b)**

### 2.3.2 Low Level Software Design



**Fig. 2.3.2 (a)**

****

**Fig. 2.3.2(b)**

# Chapter 3

# Implementation

## 3.1: Input Implementation Software

### 3.1.1: Data Collection

The dataset was taken from [11] which is distributed under CC-BY-4.0 license. It consists of 200 piano only songs in midi format classified into jazz and classical with an average length of 4 minutes each. For generating input to the model, each midi file was aligned to a specific time interval and converted into a 88x88 size matrix. This matrix representation essentially indicates that 88 time steps were taken along 88 pitches that are possible in a piano. Each entry in the matrix represents the velocity of the particular note at every timestep.

### 3.1.2: Input Software Algorithm

### 3.1.2.1: Spectrogram Preprocessing :

A spectrogram is a 2 D representation of the frequencies of a 1D signal as it varies with time.SPectrograms are generally viewed as a 1xT image of F channels. Spectrograms finds its use in fields such as music automation , sonar and speech processing . To convert an audio file to a spectrogram image we use short time fourier transforms. We apply fourier transforms of the 1D audio signal and convert them spatial domain to frequency domain. To enable ease of computing we use Fast Fourier transform which is a computational faster implementation of fourier transform. Once the spectrogram images are formed we feed them to the analysis stage.

## 3.2 Implementation of Individual Components

### 3.2.1 GAN architecture:

GAN architecture makes use of two components,namely, a generator and a discriminator. The generative model is pitted against an adversary, which is a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution[10]. In our work, the decoder from the variational autoencoder described above and as depicted in fig 1.1 plays the part of generator for the GAN. The music generated by the decoder is then passed through a discriminator in order to determine whether or not the produced resultant music is as per the required genre characteristics. The discriminator used is composed of conv2D and dropout layers. Since the generator tries to produce music such that the discriminator may classify as that of the required genre and the discriminator tries to improve its ability to decide whether an input to it was generated by the generator or it was an actual music from the required genre, they supplement each other in increasing the model accuracy to produce music in the target genre with the content from the source genre.

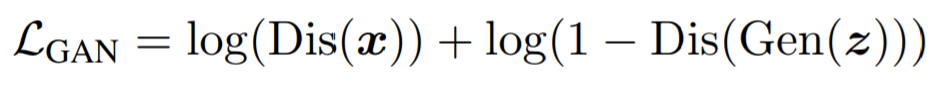
### 3.2.2 VAE-GAN Training :

The training of both VAE and GAN was done simultaneously. Therefore, multiple losses incurred at various stages of this pipeline were considered for optimization using an Adam optimizer at each stage.

The variational autoencoder loss function was approximated as a negative sum of the reconstruction error and a prior term for regularization.

The reconstruction error calculated here is the cross entropy reconstruction loss as :

The GAN consists of two networks: the generator network Gen(z) maps latents z to data space while the discriminator network assigns probability y = Dis(x) ∈ [0, 1] that x is an actual training sample and probability 1 − y that x is generated by the model through x = Gen(z) with z ∼ p(z).

The losses for the GAN are calculated below where both the generator and discriminator use sigmoid cross entropy to calculate their individual losses. The GAN loss is a combination of both these losses incurred. 

A very important role played in the quality of training and speed of convergence is the use of dynamic learning rates for both the generator as well as discriminator. With each epoch, the learning rate is modified based on the losses incurred in the previous epoch.

The resultant output is a 88x88 array that needs to be converted back into its midi representation using the pitch and volume information encoded in the matrix at various timesteps. This post processing completes the procedure for VAE-GAN based music style transfer.

### 

### 3.2.3 Spectrogram Implementation:

We propose a spectrogram analysis method which combines the information present in a spectrogram of a 1D signal with the famous style transfer method proposed by Leon Gatys[1] *et al.* for images.The success of neural art transfer for image related tasks has been to start with two images a content image and a style image and then optimize the input signals starting with random noise to take the features of interest from both the content and style images. Here we employ a similar procedure where we take the spectrogram images of two audio signals and starting with a random noise we transform it to a combination of the two input audio signals.To extract features from the spectrogram images we require a neural network that can understand the complex patterns of the spectrogram image. We first experimented with VGG-19 a state of the art Image classification network that was trained on the imagenet dataset. Being trained on image datasets we assume that the network will easily be able to extract important features from the image. But as spectrogram images are not general RGB images and 3x3 convolutions are not well suited for our 1D problem. So we move to a second method that uses 1D convolutions. Here we can either use a pretrained network that uses 1d Convolution or a network whose weights have been randomly initialized. Analysis of both the approaches shows that the random weights implementation has the same effects as the pretrained models. So we further move with the randomly initialized weights approach for image feature extraction. The features are extracted from both the input audio file spectrograms and they are compared with the features from the random noise. The loss is calculated from the loss functions shown in figure 2. Then using L-BFGS we update the weights(random noise vector) such that the loss is minimized. After the algorithm has converged the resultant spectrogram image now represents the combination of the input spectrograms.

### 3.2.4 Web Interface Implementation:

The front end of the web interface which consists of web pages allowing users to upload audio files and play the resultant audio files were implemented using HTML, CSS and Bootstrap.

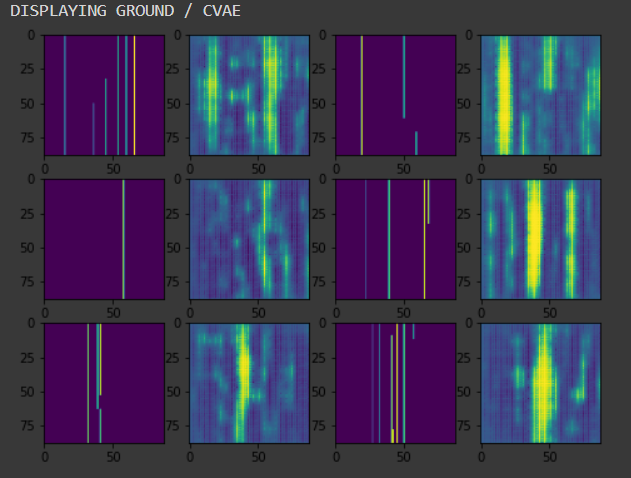
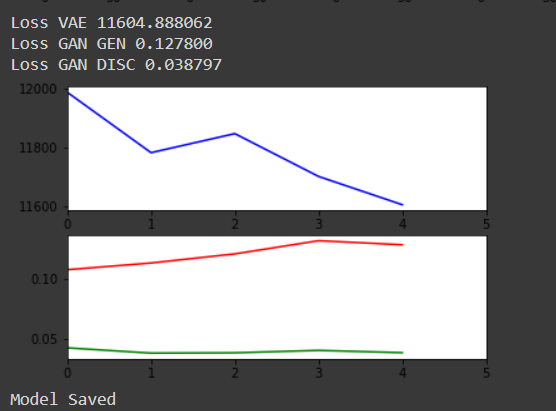
The back end processing of inputs and delivery of outputs was implemented using the python framework Flask. Flask was enabled and run on Google Colaboratory using the package ngrok. The flask framework takes the input and stores it in the local Colaboratory directory and onto the drive mounted. The models take as inputs these files and produce the resultant files which too are stored in the drive and local directory. The web page accesses these files from the local directory and provides the user an interface to play these files with suitable controls.

# Chapter 4:

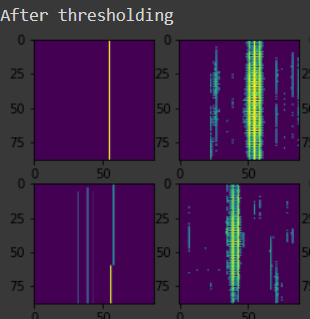
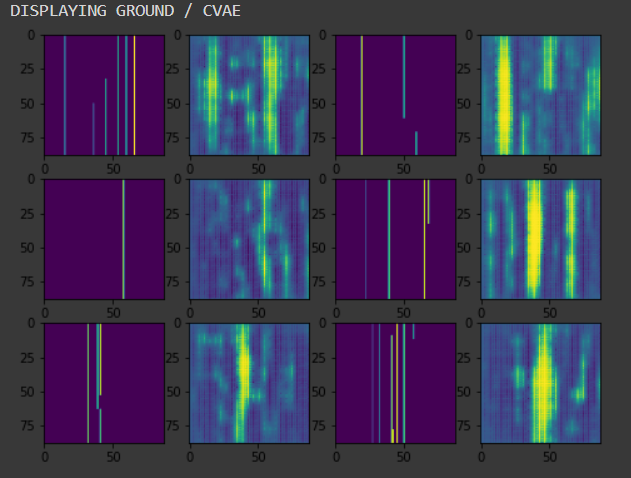
## 4.1 : Results and observations

### 4.1.1 Gan Results:

The proposed VAE-GAN model was evaluated by passing the input and output music through a music classifier and the change in genre was clearly noticed in favour of the target genre. However, the resultant music files were found to be highly noisy and it can be said that the style transfer occurred at the expense of conserving content information. This ill effect however can be overcome by training over more number of epochs and better control over the dynamic learning rate. The noise observed in the model outputs also can be credited to the fact that the only information considered for style understanding was pitch and volume at each timestep. Other composition aspects were not taken into account. To reduce the noise in output, the model makes use of a thresholding function which reduces the noise considerably.

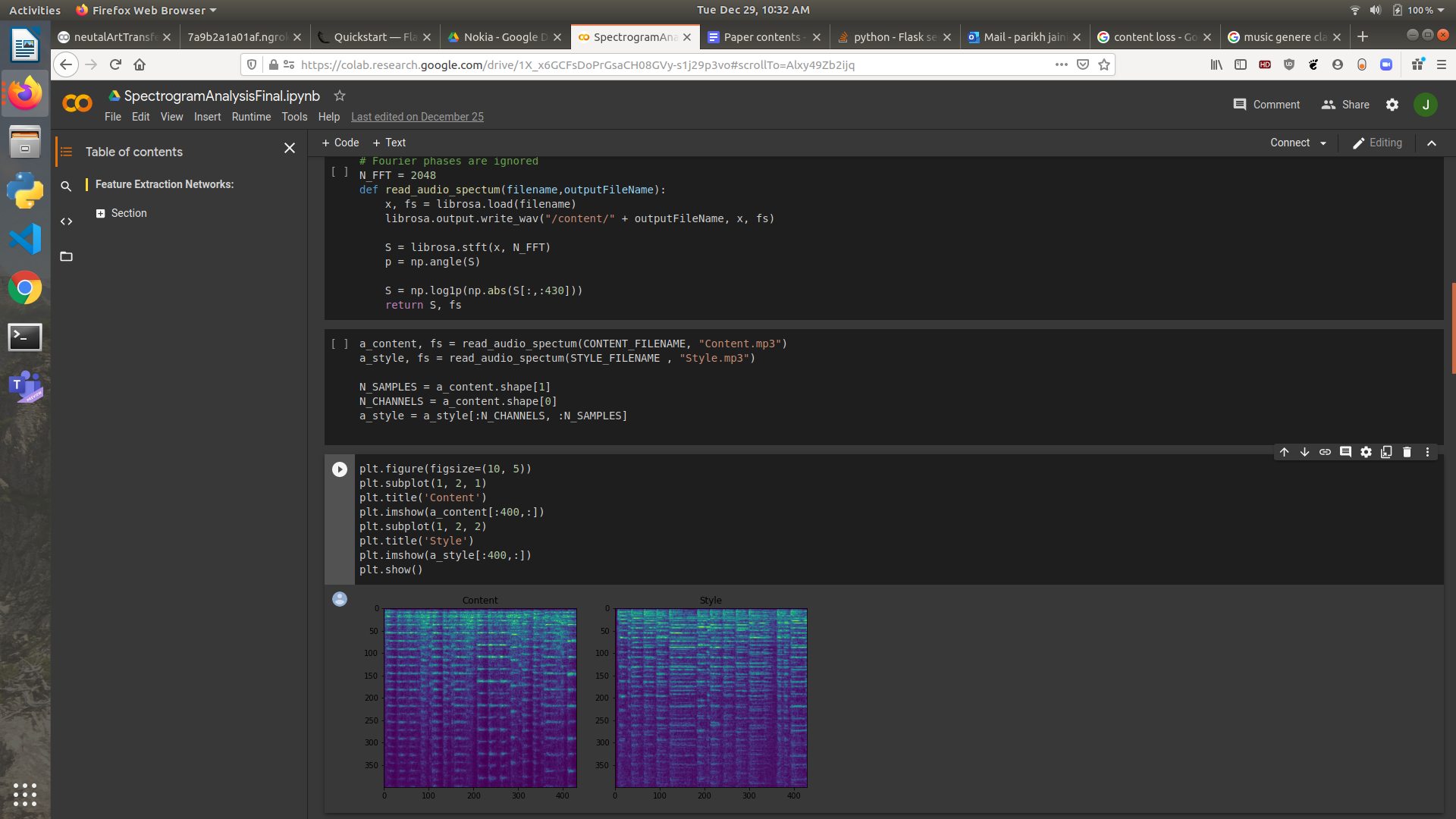
**Fig. 4.1.1(a) Fig. 4.1.1(b)**

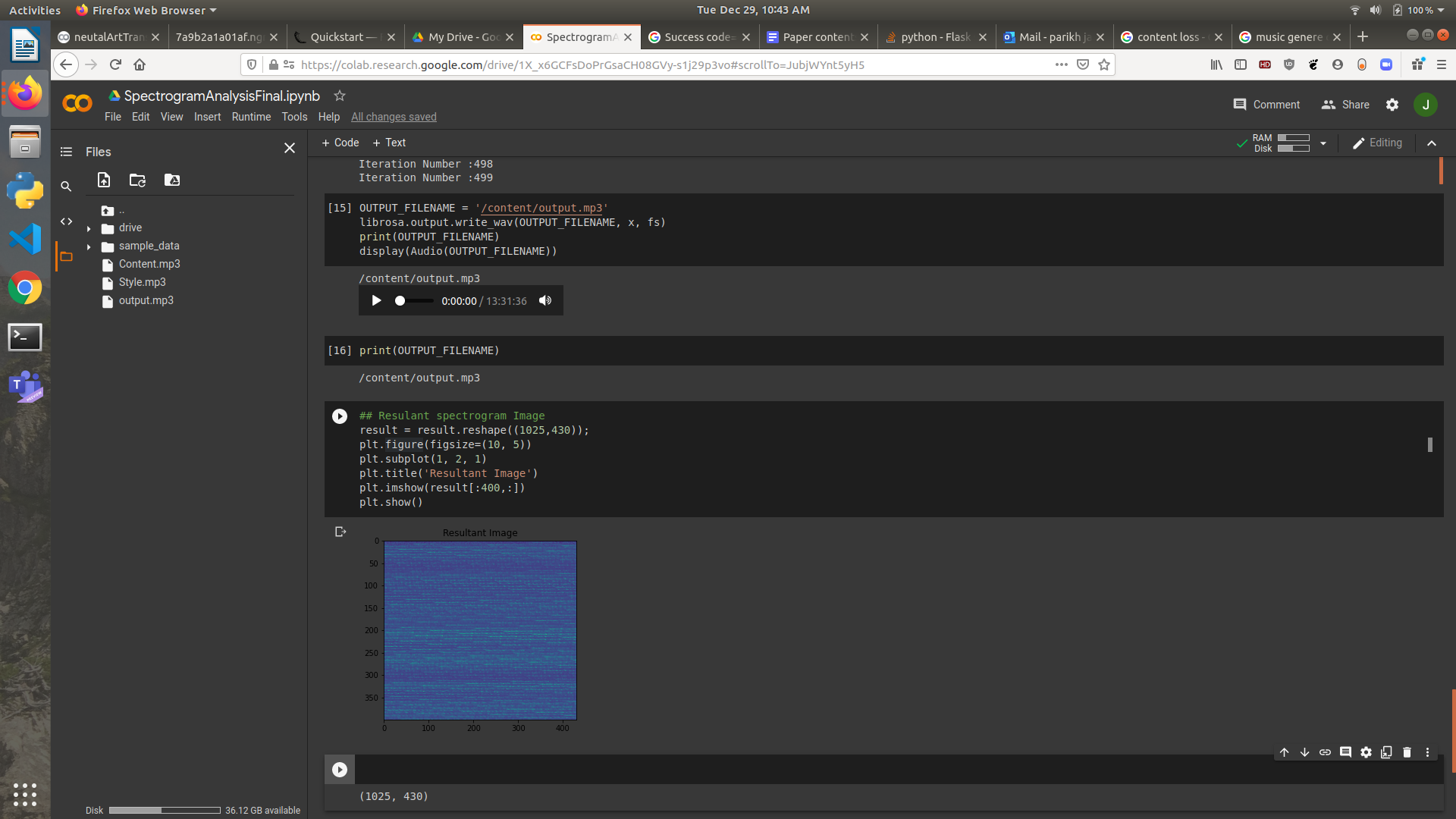
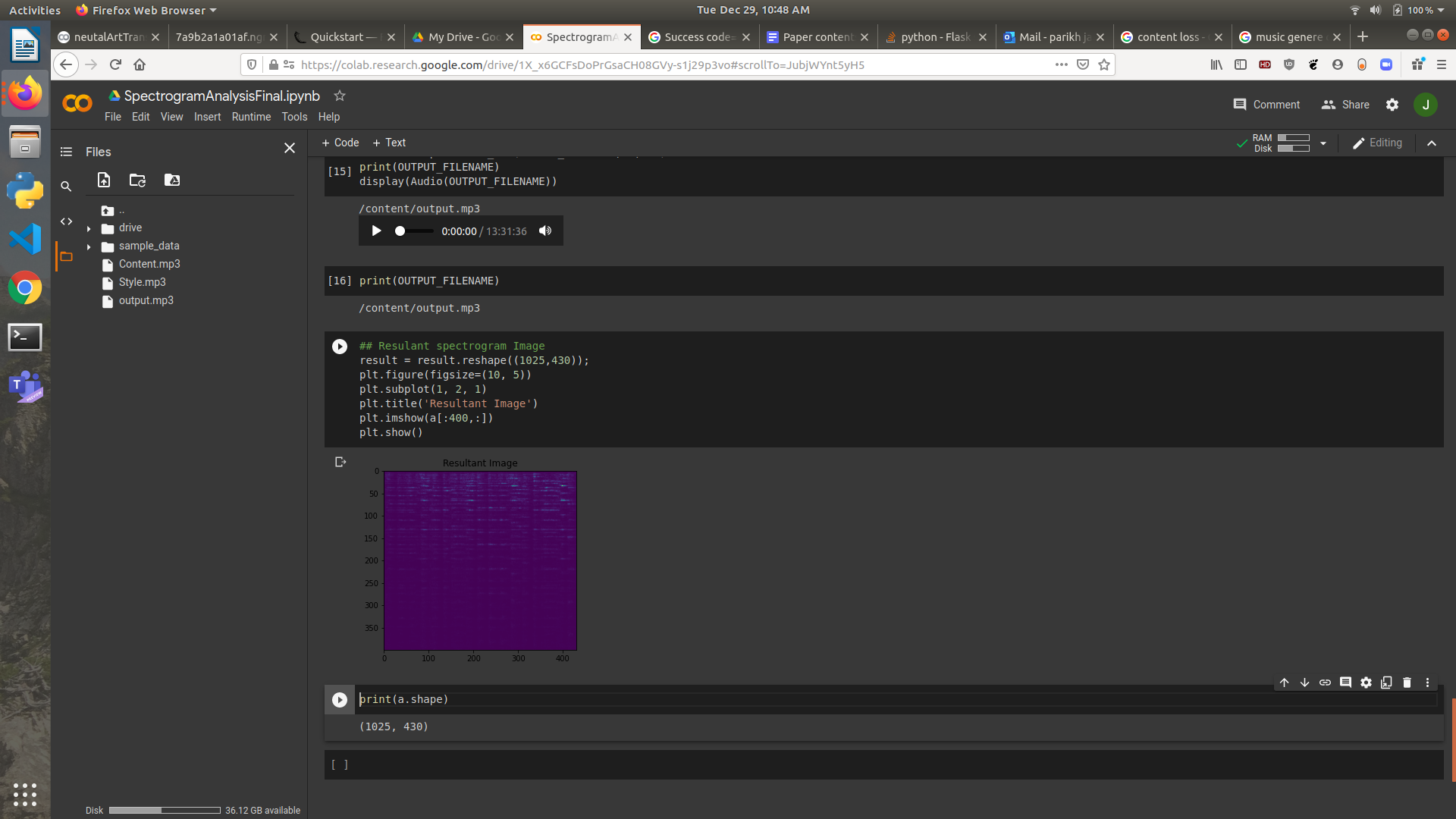
**Fig. 4.1.1**

### 4.1.2 Spectrogram Results

We evaluated the proposed spectrogram analysis approach using a music genre classifier that was trained on GTZAN . Figure 3 shows the different spectrogram images that were resultant from the VGG model and the 1D convolution models.As can be observed the output from the VGG net is extremely noisy.This is mainly due to the fact that VGG is trained for image description . The shallow net architecture with randomly assigned filters provides the best result. TO better understand the results we passed the input audio files and the resultant audio file to the music genre classifier. What we observed is that the output obtained from the VGG model was classified to a genre which is neither of the input audio genres.This clearly proves that the resultant from VGG is noisy. We apply the same classifier to the resultant audio file of the shallow network. What we observed is that the output genre is a mixture of the input genres which is the expected result.One can also hear the outputs from the shallow networks which seem to place the style texture most precisely . Our tests confirm that the use of random shallow networks with L-BFGS has provided smoother results compared to gradient descent approach.

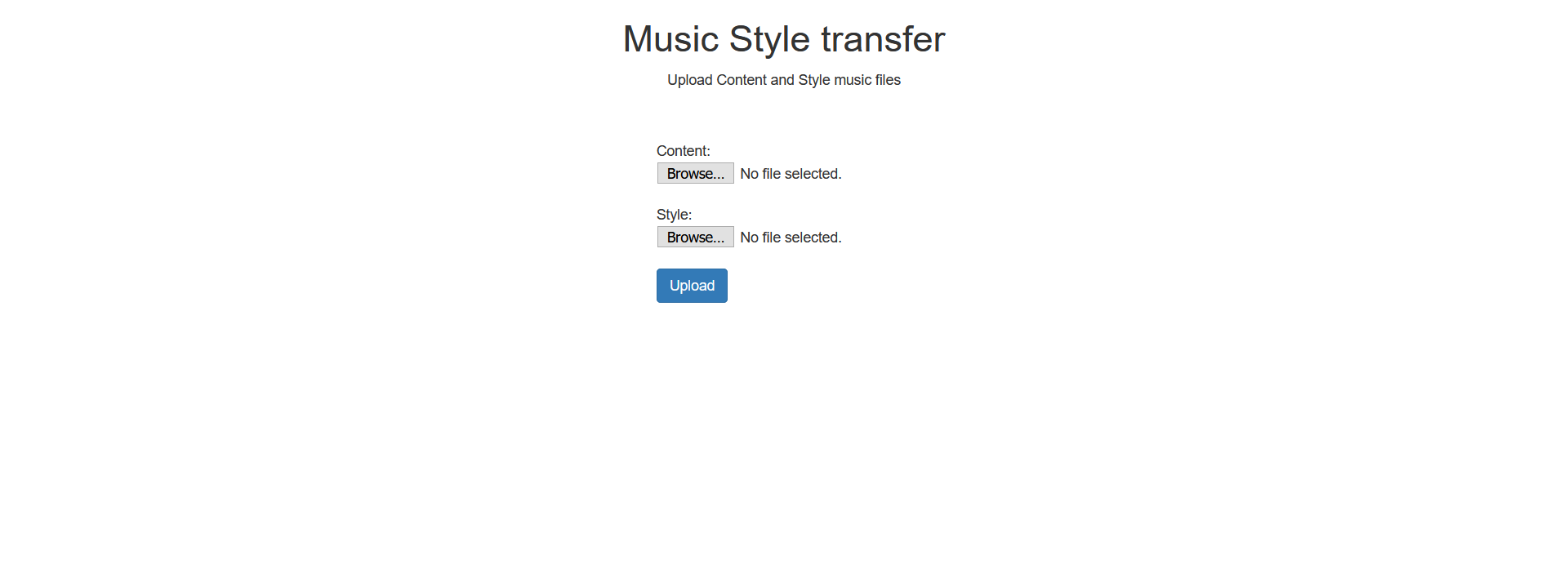


**Fig. 4.1.2(a)**

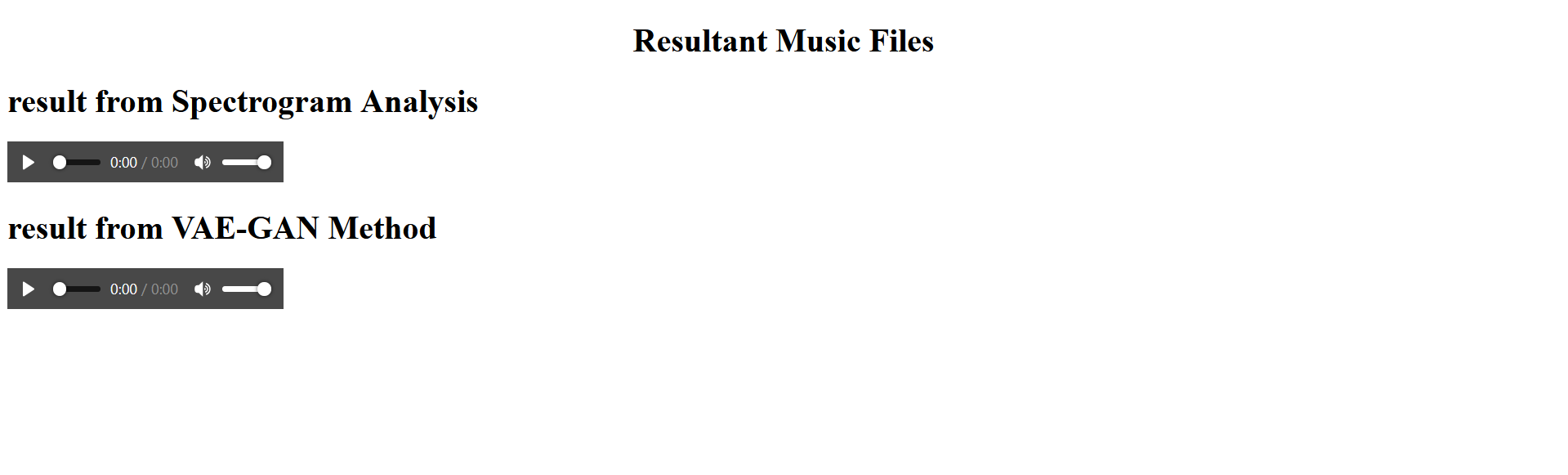
 

**Fig. 4.1.2(b)**

### 4.1.3 Interface Results:



**Fig. 4.1.3(a)**

****

**Fig. 4.1.3(b)**

# Chapter 5

# 5.1 CONCLUSION

In this work we proposed two methods for audio style transfer.One that used generative adversarial networks to understand the intricacies of single track piano music, and the spectrogram analysis model that interpreted 1D signals in frequency domain over multitrack music as well. The experiments showed that the resultant audio files from the GAN model were more noisy than the spectrogram files but showed prospects for a much greater improvement in quality over a slightly more fine tuned training from the aspect of learning rates than the spectrogram results.By comparing the results of both the models with a music genre classifier we come to a conclusion that generative models outperform spectrogram models. However, spectrogram analysis provides us with an interesting insight into the image representations of audio files.Our research shows that spectrogram analysis has also paved the way for using pre trained networks which were meant for image related tasks to be used on audio files. This research topic is very unique and at this stage numerous other relevant sound texture models should be further explored in future work.

## 5.2 Future Work

The noise observed in the model outputs also can be credited to the fact that the only information considered for style understanding was pitch and volume at each timestep. Other composition aspects were not taken into account. To reduce the noise in output, the model makes use of a thresholding function which reduces the noise considerably.In view of these issues, the following approach to enhance performance of the VAE-GAN model is proposed. The music midi file representation into the matrix format here only captures the velocity of every note at timesteps. However, efforts can be made to encode more relevant information in the matrix concerning the pitch transitions and melody. In addition to this, our model was trained at random pieces of music files of 88 steps each. This could have resulted in scenarios where the number of notes being played are too less or is filled with silence which in turn contributed to the performance degradation.

In the spectrogram analysis model what we observe is that a randomly initialized neural network outperformed other approaches.We can experiment with different architectures and further improve the accuracy of our network.We also plan to execute the model on a GPU so the execution time will be considerably less than current execution times. We can further improve our work by considering the improvements made by Facebook in this field and learning from their WaveNet architecture which is used for Universal voice conversion. So we can expand the application of our work to not only music style transfer but also to convert one human voice to another.

# References

[1] Gatys, Leon A., et al. “A Neural Algorithm of Artistic Style.” ArXiv:1508.06576 [Cs, q-Bio], Sept. 2015. arXiv.org, http://arxiv.org/abs/1508.06576.

[2] Y. Li, T. Zhang, X. Han and Y. Qi, "Image Style Transfer in Deep Learning Networks," 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, 2018, pp. 660-664, doi: 10.1109/ICSAI.2018.8599501.

[3] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, July 2002, doi: 10.1109/TSA.2002.800560.

[4] Wundervald, Bruna D., and Walmes M. Zeviani. “Machine Learning and Chord Based Feature Engineering for Genre Prediction in Popular Brazilian Music.” ArXiv:1902.03283 [Cs, Eess, Stat], Feb. 2019. arXiv.org, http://arxiv.org/abs/1902.03283.

[5]Dai, Shuqi, et al. “Music Style Transfer: A Position Paper.” ArXiv:1803.06841 [Cs, Eess], July 2018. arXiv.org, http://arxiv.org/abs/1803.06841.

[6] Brunner, Gino, et al. “MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer.” ArXiv:1809.07600 [Cs, Eess, Stat], Sept. 2018. arXiv.org, http://arxiv.org/abs/1809.07600.

[7] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman. A universal music translation network. CoRR, abs/1805.07848, 2018.

[8] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, pages 1068–1077, 2017.

[9] Yamshchikov, Ivan P., and Alexey Tikhonov. “Music Generation with Variational Recurrent Autoencoder Supported by History.” ArXiv:1705.05458 [Cs], Nov. 2018. arXiv.org, http://arxiv.org/abs/1705.05458.

[10]Goodfellow, Ian J., et al. “Generative Adversarial Networks.” ArXiv:1406.2661 [Cs, Stat], 1, June 2014. arXiv.org, http://arxiv.org/abs/1406.266