AUDIO STYLE TRANSFER LEARNING

A PROJECT REPORT

Submitted by

BANGARU MOHNISH [Reg No: RA1711003011397]
SHREYA PRAKASH [Reg No: RA1711003011238]

Under the
Guidance
of

Mrs. M. HARINI

(Associate Professor, Department of Computer Science and Engineering)

In Partial Fulfilment of the Requirements
for the
Degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2021

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B.Tech project report titled "**AUDIO STYLE TRANSFER LEARNING"** is the bonafide work of **Mr. BANGARU MOHNISH and Miss SHREYA PRAKASH** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

<<Signature of the Supervisor>>
<<Signature>>

Mrs. M. HARINI
SUPERVISOR
Assistant Professor
Department of Computer Science and Engineering

Dr. B. AMUTHA
HEAD OF THE DEPARTMENT
Department of Computer Science and Engineering

Signature of the Internal Examiner

Signature of the External Examiner

Department of Computer Science and Engineering

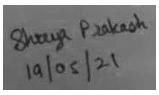
SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : Bachelor of Technology/ Computer Science and Engineering

Student Name : Shreya Prakash & Bangaru Mohnish  

Registration Number : RA1711003011238 & RA1711003011397

Title of Work : Audio Style Transfer Learning

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ACKNOWLEDGEMENT

We express our humble gratitude to **C. Muthamizhchelvan**, Vice Chancellor (I/C), SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for his invaluable support.
We wish to thank **Dr. B. Amutha**, Professor & Head, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor **Dr. P. Vishalakshi,** Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for their great support at all the stages of project work. We would like to convey our thanks to our Panel Head, **[Panel head name],** [Designation], Department of Computer Science and Engineering, SRM Institute of Science and Technology, for his / her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisors**, Dr. Godfrey Winster,** Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology and **Mrs. Aswathy. K. Cherian,** Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Mrs. M. Harini,** Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his/her mentorship. She provided me the freedom and support to explore the research topics of my interest. Her passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

Shreya Prakash

Bangaru Mohnish

ABSTRACT

Neural Style Transfer refers to a set of algorithms that manipulate and modify images . It requires two images, content image and a style image. Style image can be any popular painting. The job of neural style transfer is to manipulate them in a way that the result image looks like the content image but “painted” in the style of style image. It is a concept which has been applied to image domain mostly with the example of creating a Van Gogh painting from any given input image.

Aim of this project is to understand this concept on audio domain and implement it. We have consolidated works related to how the style of an audio or image can be transferred to another content audio or image. We have studied various Neural Style Transfer Algorithms for images(like Gatys et al, Ulyanov et al, Li and Wand, Chen and Schmidt, etc). Many researchers have used Gatys et al as their primary algorithm for building their own model as their main approach was to extract high level feature representations from an image. The content image is replaced with the style image while staying loyal to the semantic features of content. Both of these goals can be achieved by viewing this as an optimization problem.

NST for audio signals include deep neural networks which require pretraining. Many works suggest replacing the images used in NST algorithms with image representations of audio and carry out the same tasks used in image domain. The main problem here is most of the images are well defined and they have very less abstract content whereas image representations of audio signals are mostly abstract. Hence, it is crucial to find the correct audio representations and model architecture suitable for audio signals.

Present systems use pretrained networks like VGGnet inspired by NST for image domain. Extraction of key features from paintings include brush strokes and swirls whereas audio signals have rhythm and tempo. CNN models suitable for images might not work with the same efficiency in audio domain. Images contain contiguous pixels whereas audio signals are not contiguous in the frequency dimension. It is still crucial to understand image CNN models to work on audio CNN models as the former has been developed more and it will provide an insight in creating the same for audio domain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

S.NO	SYMBOL	FULL FORM
1	CNN	Convolutional Neural Network
2	STFT	Short Term Fourier Transform
3	NN	Neural Network
4	NST	Neural Style Transfer
5	ANN	Artificial Neural Network
6	1D-CNN	One Dimensional Convolutional Neural Network

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Music and art play important roles in our lives. Neural style transfer enables any person to become an artist. People can transfer the style of their chosen artist's audio/image to another content audio/image and generate a new output. Other than music, Neural Style Transfer can be applied on audio to enable people with speech problems become an orator. We need to understand this concept on images before we experiment it on audio.

Currently, Neural Style Transfer is hugely popular among artists across the globe. There already applications implementing NST for image domain like "Prisma" and "DeepArt". The user feeds in their own photogram then chooses a style of painting. The app generates a picture which depicts how the chosen artwork's painter would have painted the user supplied photograph.

Leon Gatys et al was the first one to publish his works on Neural Style Transfer in 2015. All the researchers after him have been following his basic approach towards stylizing pictures. The model used by Gatys was VGG-19 architecture and it was pretrained on ImageNet dataset.

The aim of NST algorithms is to generate an output image that shows the content of "content" image applied with the style of "style" image.

Audio processing techniques have to be applied to convert audio to spectrogram and vice versa. A spectrogram is a pictorial representation of a one-dimensional audio signal. Artificial Neural Networks have to be employed to perform the blending of both content and style audios. Convolutional Neural network is a type of deep neural network popular for analyzing images. Chosen content and style audios have to be converted to images then Neural Style Transfer can be performed on them. High level feature maps have to be extracted from these spectrograms. A loss function is computed from both content and style feature maps. This loss function is iteratively optimized to create the desired output spectrogram. The output image has to be converted back to audio.

1.2 PURPOSE

Neural Style Transfer on audio will be mainly useful in the music industry. As it just requires two audios, content and style, one can create new music just by Neural Style Transfer. Not just professionals but even amateurs can use our tool to generate different new kinds of melody. Neural Style Transfer is already popular on images. There is an app called prisma which takes in an input image and paints it in the style of the user's chosen style image.

NST saves time and money as one does not have to learn or buy different instruments to make music. Musicians have to book recording studios and hire professionals for guitar, piano, etc.
With our tool, one just needs to experiment on content and style audio signals to create their own music.

1.3 SCOPE

An app can be created for NST on audio just like how prisma was developed for images. It can be a fun activity for children and adults to generate their own music.

It can be further developed to help people with speech problems become an orator. For example, if a person stammers, this tool can be improved to bring out their voice properly. There are many conditions where this tool will be helpful like Aphasia(Communication disorder that makes it hard to use words), Dyslexia(They have difficulty breaking down words into simple sounds), Spoonerism(Mixes up sounds of words), etc.

As of now, little to no work has been done on NST on audio. It has a wide scope in the music and film industry. It will cut their instrument and studio costs for creating raw fresh music.

CHAPTER 2

LITERATURE REVIEW

2.1 Universal Style Transfer via Feature Transforms (2017)

Authored by: Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, Ming-Hsuan Yang

The study demonstrates a model to transfer styles to content images and produce high quality output. Feed-forward based methods are able to blend content and style images but they are unable to work on unseen styles or low visual quality. The model presented in this paper deals with such problems and it does not need to train on any pre-defined styles.

The main problem is extracting the true representation of the style and then blend it in the content image. Work by Gatys et al. prove that the Gram matrix or covariance matrix generated by a trained deep neural network has the desired ability of extracting visual styles.

First, feature extraction is done by using the VGG-19 network (19 layered CNN) , then these features are inverted to the original image by a symmetric decoder (image reconstruction) next WCT (whitening and coloring transforms) is applied. The blended features are further fed forward into the decoder layers in order to generate the stylized image. This algorithm does not need to train on style images unlike the old algorithms. Extraction of feature gram matrices and application to the content features through WCT is the way to deal with new styles.

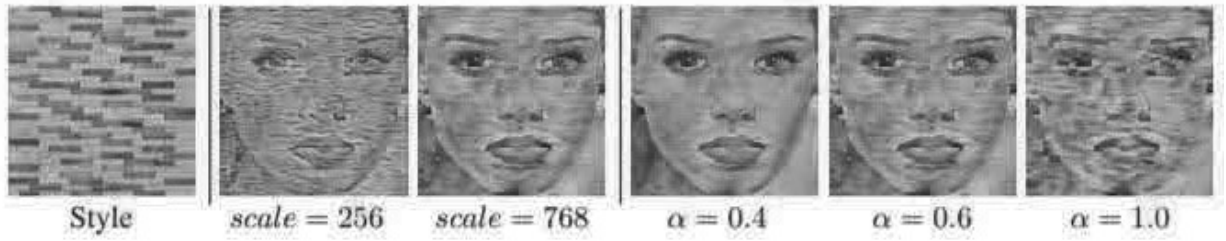
The efficiency of WCT is shown by comparing it with a popular feature adjustment technique, HM (histogram matching). It is evident in their output images that the HM technique does the job but could not capture prominent visual patterns like patterns that are torn into several bits and local structures are not shown properly. On the other hand, WCT technique is able to understand patterns that represent the style image better. This is because the HM technique is unable to see the correlations among features, which is what the gram matrix is made for.

Table 1: Differences between our approach and other methods.

	Chen et al. [3]	Huang et al. [15]	TNet [27]	DeepArt [9]	Ours
Arbitrary	√	√	×	√	√
Efficient	√	√	√	×	√
Learning-free	×	×	×	√	√

Table 2: Quantitative comparisons between different stylization methods in terms of the covariance matrix difference (L_s), user preference and run-time, tested on images of size 256×256 and a 12GB TITAN X.

	Chen et al. [3]	Huang et al. [15]	TNet [27]	Gatys et al. [9]	Ours
$\log(L_s)$	7.4	7.0	6.8	6.7	**6.3**
Preference/%	15.7	24.9	12.7	16.4	**30.3**
Time/sec	2.1	0.20	0.18	21.2	**0.83**



As art is highly subjective, they conducted a user study to evaluate all the different methods. They used five content and thirty style images to obtain one hundred and fifty different result output images on each pair of content/style for every technique. Then randomly chose fifteen style images for every

subject to analyze. They showed stylized images by the compared techniques together in random order. People are asked to vote their favorite for each style, and their algorithm was the most popular choice.

As art is highly subjective, they conducted a user study to evaluate all the different methods. They used five content and thirty style images to obtain one hundred and fifty different result output images on each pair of content/style for every technique. Then randomly chose fifteen style images for every

subject to analyze. They showed stylized images by the compared techniques together in random order. People are asked to vote their favorite for each style. They collected feedback and it showed that their algorithm received the most votes for better stylized results

2.2 Image Style Transfer Using Convolutional Neural Networks (2016)

Authored by: Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

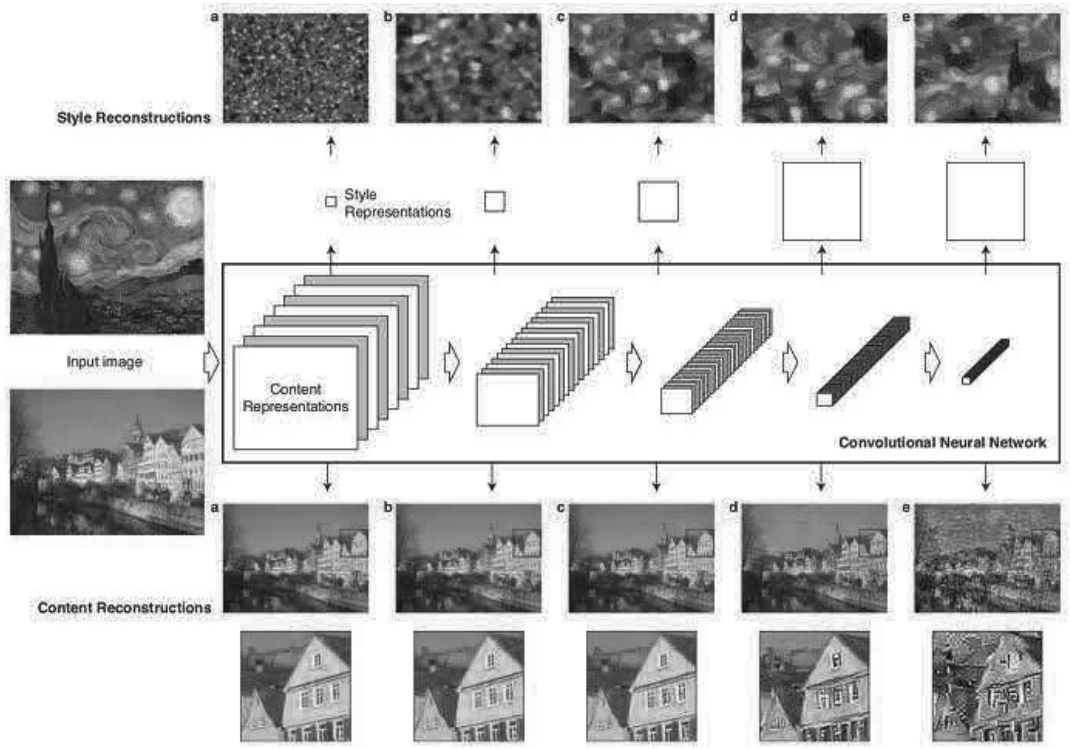
This study demonstrates an algorithm that is able to separate and recombine the style and content images. It generates new images of high visual quality that blends the content of an input image with the style of various popular paintings. Their results show that the Convolutional Neural Networks are able to perform high level image synthesis and manipulation. A style transfer algorithm has to obtain the effective representation from the target image (e.g a landscape). It has to further perform a texture synthesis to render the target image in the style of the artwork.

The transfer process is carried out in two modules one for content and the other for style.

The source images are passed through a convolutional neural network with custom filters to extract local and global features of the content image and similarly for the style image, we design a feature space to capture the texture
information it consists of relations between the different filter responses.

They match the content representation of a photograph depicting the riverfront of the Neckar river in T¨ubingen,
The total loss is computed with the weighted sum of the content loss and the style loss, this loss is further minimized using an optimizer.

FIGURE 1:The flow diagram of the proposed system:



The key finding of this paper is that the representations of content and style in the Convolutional Neural Network are well separable to independently to produce new, perceptually meaningful Images.

2.3 Neural Style Transfer: A Review (2018)
Authored by: Yongcheng Jing, YezhouYang, ZunleiFeng, JingwenYe, Mingli Song

This study presents an overview of Neural Style Transfer Models. It evaluates and compares various algorithms used for NST. Finally it discusses about the several applications and scope for future development. The authors help us choose the optimal method for style transfer. It shows the numerical results of every method helping in accurate comparison. Gatys et al. first demonstrated the use of Convolutional Neural Network to regenerate popular artwork styles on natural images.

The study focuses on image-based artistic rendering (IB-AR)

Stroke-Based Rendering(SBR)- It denotes a procedure of putting strokes on a canvas digitally to create a painting with a particular style. It is successful in showing a particular

style but it can only follow one style. It is perfect for oil paintings, water colours and sketches.

Region-Based Techniques- It simplifies shape rendering effects by placing canonical shapes instead of regions. It can only follow one style like SBR.
Example-Based Rendering- It maps between a content images pair and target stylized images pair in a supervised way. They can only capture low level features. Hence they are not ideal.

Image Processing and Filtering- It simplifies and abstracts images. They are effective but cannot adopt different styles.

All these limitations of IB-AR techniques paved way for the rise of Neural Style Transfer. It is the procedure of extracting style from an image then using it for content image reconstruction.

Here are 5 different NST algorithms tried on six different content and style images.

FIGURE 2: This is the qualitative comparison.

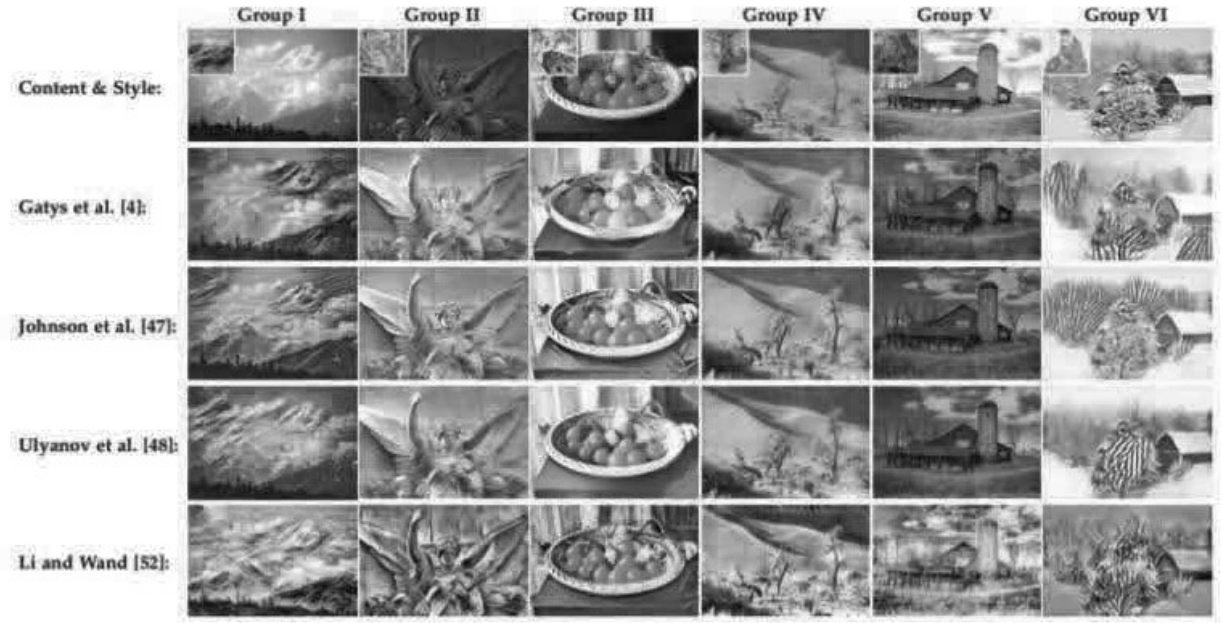


TABLE 3: Average speed comparison for sizes in pixels

Methods	Time(s)			Styles/Model
	256 × 256	512 × 512	1024 × 1024	
Gatys et al. [10]	14.32	51.19	200.3	∞
Johnson et al. [47]	0.014	0.045	0.166	1
Ulyanov et al. [48]	0.022	0.047	0.145	1
Li and Wand [52]	0.015	0.055	0.229	1
Zhang and Dana [56]	0.019 (**0.039**)	0.059 (**0.133**)	0.230 (**0.533**)	$k(k \in Z^+)$
Li et al. [55]	0.017	0.064	0.254	$k(k \in Z^+)$
Chen and Schmidt [57]	0.123 (**0.130**)	1.495 (**1.520**)	–	∞
Huang and Belongie [51]	0.026 (**0.037**)	0.095 (**0.137**)	0.382 (**0.552**)	∞
Li et al. [59]	0.620	1.139	2.947	∞

TABLE 4: Comparison for efficiency, arbitrary style and learning-free

Methods	E	AS	LF
Gatys et al. [4]	×	√	√
Ulyanov et al. [47]	√	×	×
Johnson et al. [50]	√	×	×
Li and Wand [52]	√	×	×
Dumoulin et al. [53]	√	×	×
Chen et al. [54]	√	×	×
Li et al. [55]	√	×	×
Zhang and Dana [56]	√	×	×
Chen and Schmidt [57]	√	√	×
Ghiasi et al. [58]	√	√	×
Huang and Belongie [51]	√	√	×
Li et al. [59]	√	√	√

Prisma is an app for converting a natural image into a painting in a style of famous artwork. It uses Neural Style Transfer and it has achieved a lot of popularity. Ostagram is a similar app providing faster speed.
NST can be useful for painters, fashion designers and artists who want to experiment with their ideas.
Creation of animations will become simpler and easier if NST is used for it.

NST is rising but it still has a long way to go. There is a need to refine the new techniques, looking for properly fitting different kinds of styles. It performs well on paintings but for some styles, it produces irregular output due to image construction from CNN. It works on natural content images perfectly but fails when it takes an abstract image as content.

2.4. Deep Photo Style Transfer (2017)
Authored by: Fujun Luan, Sylvain Paris, Eli Shechtman, Kavita Bala.

The authors of this research paper points the transfer learning concept in a new direction by applying transformations from the input to the output to be locally affine in colorspace of the image, and to express this constraint as a custom fully differentiable energy term.

Introduces a deep-learning approach that faithfully transfers style from a reference image for a wide variety of image content.
The transfer had to address the major challenges:

1. Structure Preservation
2. Semantic accuracy

The transfer of style to the image must be done in a manner in which the image doesn't lose its basic edges and the structural details of the content this is achieved by restricting the results to be photorealistic hence differentiating the Photographs from the paintings.

The Semantic accuracy plays an important role when the data being styled is a photograph such as the different objects in the picture are not mismatched during the transfer

The style transfer uses a photorealism regularization parameter in the objective function for optimization to prevent distortions.
This approach is compared against the CNNMRF

The total loss term computed in this process is governed by the following equation (CNNMRF):

$$\mathcal{L}_{\text{total}} = \sum_{\ell-1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \Gamma \sum_{\ell-1}^{L} \beta_\ell \mathcal{L}_s^\ell \qquad \text{(1a)}$$

Equation with the photorealistic regularization parameter (λ)
(New Approach):

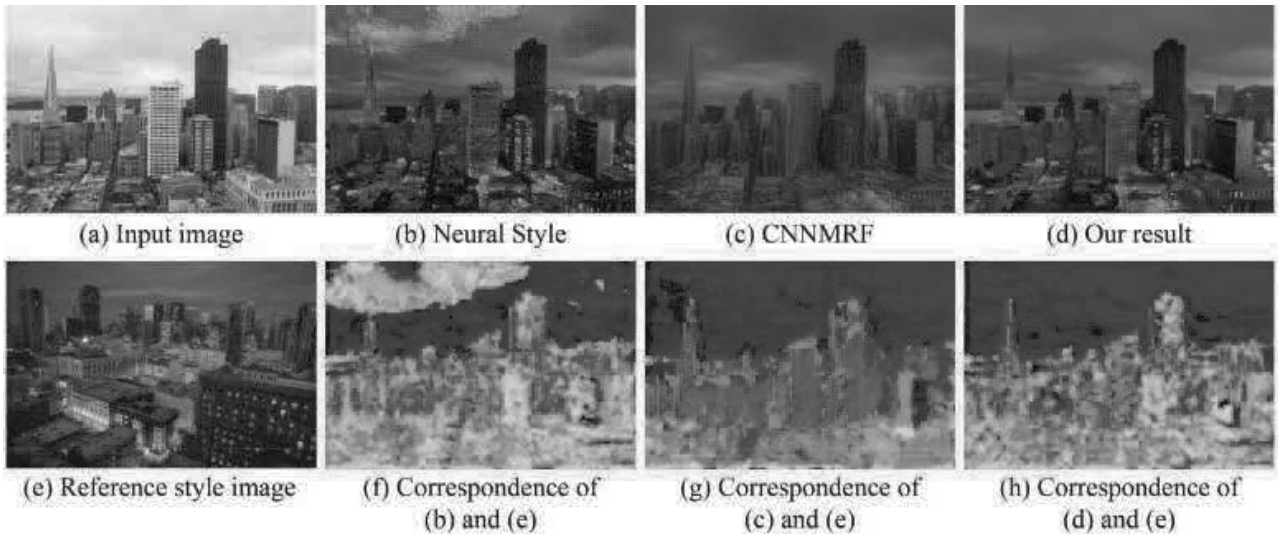
$$\mathcal{L}_{\text{total}} = \sum_{l-1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \Gamma \sum_{\ell=1}^{L} \beta_\ell \mathcal{L}_{s+}^\ell + \lambda \mathcal{L}_m \qquad (4)$$

The results achieved by using and changing the photorealistic regularization parameter:

(a) Input and Style (b) $\lambda = 1$ (c) $\lambda = 10^4$, our result (d) $\lambda = 10^8$

Figure 3: Transferring the dramatic appearance of the reference style image onto an ordinary flat shot in (a) is challenging. We produce results using our method with different λ parameters. Too small a value of λ cannot prevent distortions, and thus results in a non-photorealistic look in (b). Conversely, too large a value of λ suppresses the style to be transferred yielding a half-transferred look in (d). We found the best parameter $\lambda = 10^4$ to be the sweet spot to produce our result (c) and all the other results in this paper.

Figure 4: Comparative analysis of different methods:



(a) Input image (b) Neural Style (c) CNNMRF (d) Our result

(e) Reference style image (f) Correspondence of (b) and (e) (g) Correspondence of (c) and (e) (h) Correspondence of (d) and (e)

The Neural Style algorithm doesn't succeed in isolating the different patches of the image for the style transfer and causes distortion in the resultant image the original image CNNMRF model fails to identify the different portions of the image and causes improper or blurred resultant image, therefore adding the photorealistic regularization factor along with the Structure Preservation and the Semantic accuracy filters dramatically increases the effectiveness of the transfer.

\

2.5. Audio Style Transfer (2018)

Authored by: Eric Grinstein, Ngoc Q. K. Duong, Alexey Ozerov, Patrick P´erez

The authors of this research paper propose a flexible framework for the task, which uses a sound texture model to extract texture statistics followed by optimization and the stylized signal generation.

The authors of this research paper have approached the style transfer of audio signals using a new approach by synthesizing a custom audio texture model inspired by the human hearing mechanism.
The transfer is carried out in two stages:

1. Style extraction
2. Style transfer

The content and the style waveforms are transformed into 2D spectrogram using the Short-term Fourier Transform and the texture statistics are extracted by the sound texture model (Neural Network) after the extraction of the texture statistics the optimization algorithm is used to generate the final signal and finally the post-processing is done to convert the signal into its waveform.
This approach stands out as it does not accommodate content loss and the final signal is optimized to minimize the style loss.

Style Loss Function:

$$\mathcal{L}(\mathbf{x}; \mathbf{x}_{\text{sty}}) = \sum_{\ell \in L} \left\| \mathsf{G}_\ell(\mathbf{x}) - \mathsf{G}_\ell(\mathbf{x}_{\text{sty}}) \right\|_F^2, \qquad (1)$$

Experiments conducted by the authors point that the sound texture model based on the human auditory system and the shallow random net models produced significant results. The author also concludes that initializing the iterative optimization by the content sound, plays an important role in obtaining these results.

2.6. Neural Style Transfer for Audio Spectrograms (2018)

Authored by: Prateek Verma, Julius O. Smith

The Authors of this Research Paper aim to present a machine learning technique for style transfer on audio signals analogous to the approach of Gatys et al on images by using a convolutional neural network, and investigates the generation of spectrograms from noise

The authors approached the problem of style transfer on audio signals by conducting two fundamental experiments:

1.Imposing the style of a tuning fork on a harp
2.transferring the style of a violin to a voice

The authors explored the problem thoroughly and noted the various hyper-parameters that need to be tuned to achieve the results for the two experiments and aimed to have one parameter setting that can perform well in both the scenarios.
Such that hand tuned parameters for each scenario can be avoided.

The objective equation used:

$$X_{recon} = \text{argmin}_X \mathcal{L}_{\text{total}} = \text{argmin}_X \alpha L_c(x, x_c) + \beta L_s(x, x_s) + \gamma L_e(x_e, e_s) + \delta L_t(x_t, t_s).$$

Where X denotes the reconstructed signal, Lc is the content loss and Ls is the style loss the goal is to minimize the sum of
the weighted loss terms of both the signals.

Results:

Experiment 1:

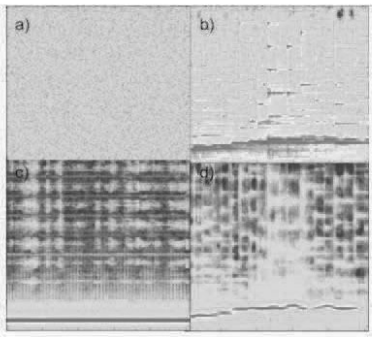


Figure 1: a) shows the Gaussian noise from which we start the input to optimize, b) Harp sound (content) c) Tuning Fork (style) and d) Neural Style transferred output with having content of harp and style of tuning fork https://youtu.be/UlwBsEigcdE

Experiment 2:

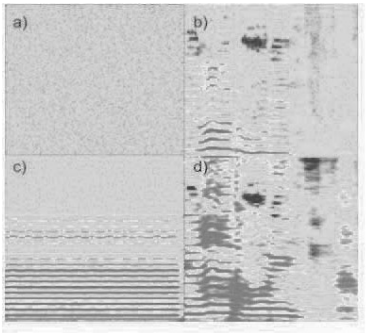


Figure 2: a) shows the Gaussian noise from which we start the input to optimize, b) Singing sound (content) c) Violin note (style) and d) Neural Style transferred output with having content of singing and style of violin. https://youtu.be/RpGBkfs24uc

The authors of the research paper have proposed a novel way to handle audio signals by considering them to be style transfer problems, by using the back propagation technique to optimize the output sound to conform to the filter outputs of a pre-trained neural network architecture. The approach proposed is very flexible and can be applied to many varieties of scenarios as discussed above.

2.7 Signal Estimation from Modified Short-Time Fourier Transform (1984)

Authored by: Daniel W. Griffin, Jae S. Lim

The authors of this research paper aim to present an algorithm to estimate a analogue signal from its modified Short-term Fourier transform.

The proposed algorithm is the iterative short-term Fourier transform it works on the concept of decreasing the ea squared error between the Short-term Fourier transform of the signal and the Modified Short-term Fourier transform of the signal.
In the time scale modification of speech.

The major computation of the proposed algorithm is the Discrete Fourier Transform.

The Authors present comparative study between the existing algorithm and the proposed algorithm

Existing algorithm: LSEE-MSTFT
Proposed algorithm: OA-MSTFT
The LSEE-MSTFT algorithm can be applied in two methods
Overlap add method

1. Iteratively

The goal of these algorithms is to minimize the mean squared distance between the STFT and the MSTFT of the analogue signal.

Distance Equation:

$$D[x(n), Y_w(mS, \omega)] = \sum_{m=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} [x_w(mS, l) - y_w(mS, l)]^2 .$$

Results:
The results have been generated for the following audio signal:

GRIFFIN AND LIM: SIGNAL ESTIMATION FROM MODIFIED STFT

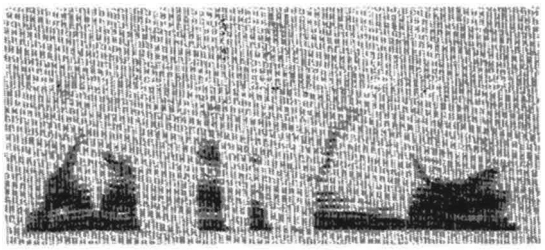


Fig. 2. STFTM of "line up at the screen door."

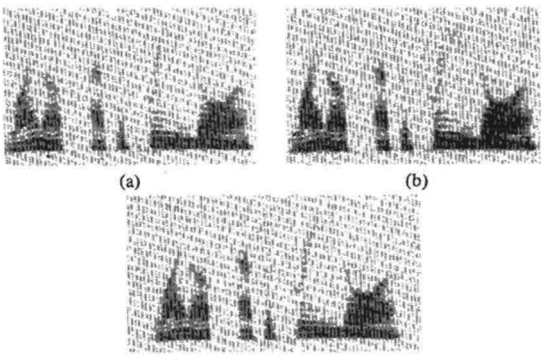


Fig. 3. (a) 128:64 time-scale compressed STFTM of original speech. (b) STFTM of LSEE-MSTFTM estimate. (c) STFTM of OA-MSTFTM estimate.

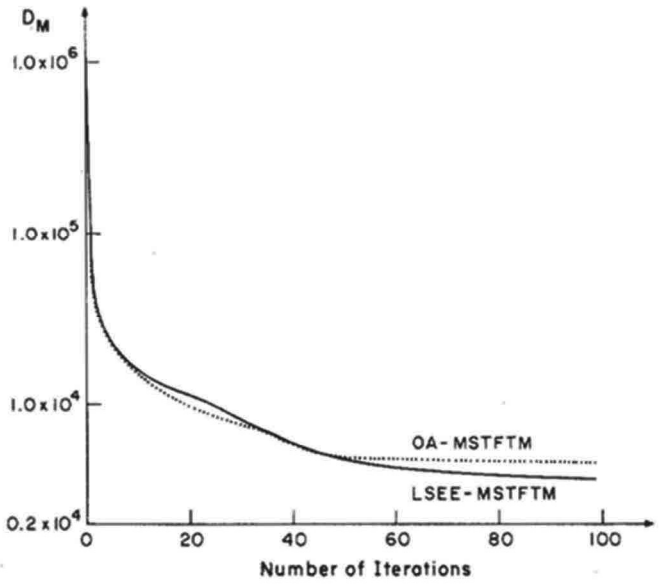


Fig. 4. D_M versus iteration number of LSEE-MSTFTM and OA-MSTFTM.

The paper presents 3 new algorithms and presents the comparative study between them as shown above
It can be inferred that even though LSEE-MSTFTM and
OA-MSTFTM had similar performance LSEE-MSTFTM has always produced minimal DM as compared to OA-MSTFTM
Over large number of iterations and can cause significant differences in different applications.

2.8 Fast Signal Reconstruction From Magnitude Stft Spectrogram Based On Spectrogram Consistency (2010)

Authored by: Jonathan Le Roux, Hirokazu Kameoka, Nobutaka Ono, Shigeki Sagayama

The Research paper presents the latest advancements on the applications of Short-term Fourier Transform in Signal Reconstruction from an altered Spectrogram.

This article reviews the concept of spectrogram consistency and the method to derive the numerical consistency criterion and proceeds to assess and investigate the performance of the fast phase estimation algorithms.

The algorithms investigated are the Griffin and Lim's iterative Short-term Fourier transform and the Fast minimization of (I)

Griffin and Lim proposed the iterative STFT algorithm, which consists in iteratively updating the phase at step k by replacing it with the phase of the STFT of its inverse STFT, while keeping the A(magnitude) fixed.

The Fast Minimization of I algorithm aims to cut the cost of computing the STFT and the Inverse STFT of a signal by taking advantage of sparseness of the magnitude array(A) and truncating A accordingly and proceeding with the phase estimation.

The experimental analysis was conducted on 100 speech samples of both male and female candidates, taken from Bagshaw's database for the length of 5 minutes and 8 portions of musical pieces of piano and guitar solos from the RWC music database for the length of 3 minutes

The authors introduced a framework to estimate the phase of the signals by studying the relation between Griffin and Lims's method and their own method the Fast minimization of (I) and developed the framework to take advantage of the sparseness of A(Magnitude array) and perform the estimation in a more efficient manner.

2.9 Adam: A Method For Stochastic Optimization (2015)

Authored by: Diederik P. Kingma, Jimmy Lei Ba

The Authors of the Research paper introduced a Stochastic Gradient optimization algorithm that is computationally efficient and has minimum memory requirement and which can be used for problems which are large in terms of parameters and data and that can be applied to methods with noise in the data or has sparse gradients.

Proposed Algorithm:

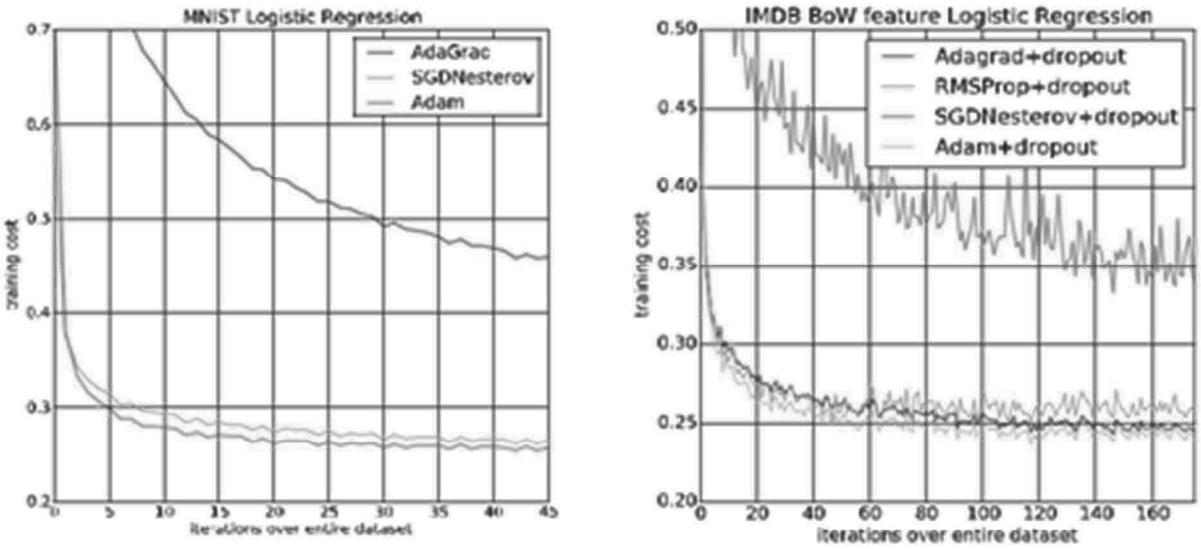
Let f be a noisy stochastic scalar function that is differentiable. The goal of the algorithm is to minimize the expected value of this function with respect to its parameters.

Pseudocode for the algorithm:

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

Results:
Performance of Adam Optimizer on MNIST and IMDB data



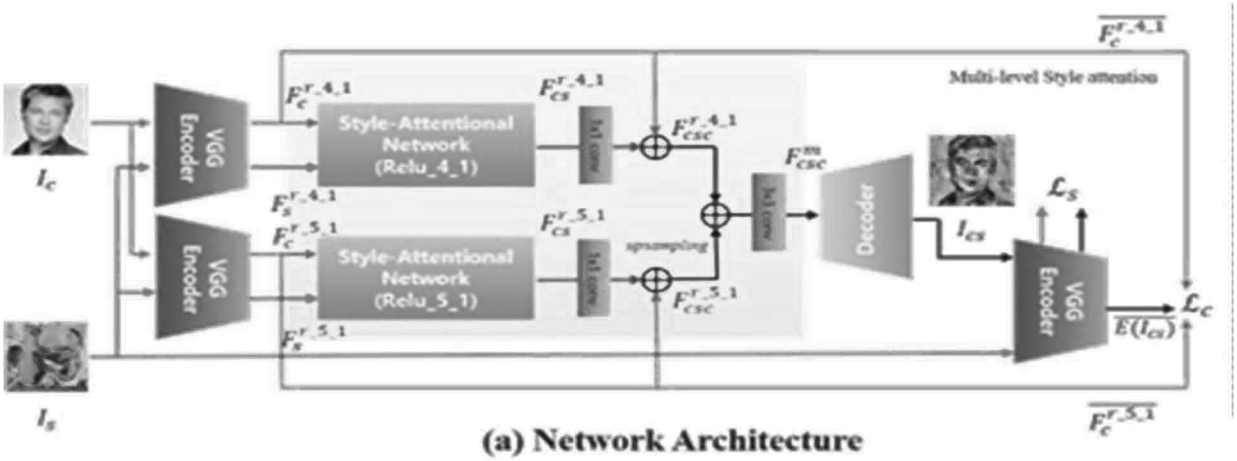
The Authors have successfully created a Stochastic Gradient Optimizer that is more efficient in regression tasks and out performs the existing optimizing methods such as AdaGrad and SGDNesterov.

2.10 Arbitrary Style Transfer with Style-Attentional Networks (2019)
Authored by: Dae Young Park, Kwang Hee Lee

The Research Paper introduces and proposes the use of a novel style attention network called the (SANet) and a new Identity loss function and multi-level feature embeddings to tackle the issue of maintaining local and global style patterns and the content structure.

The Style transfer operation proposed in the paper is composed on an encoder-decoder module and the Style-attentional module. The SANet Architecture is as follows, the encoder module and the symmetric decoder module is a pre-trained VGG-19 network.
To capture the global and the local style patterns two SANet's have been used the inputs take are the feature maps encoded from arbitrary layers of the VGG, the output is generated by combining the output feature maps. Adam optimizer was used with the earning rate of 0.0001 during the training of the model.

Network Architecture:



(a) Network Architecture

The Model was trained using MS-COCO for the content images and WikiArt for the style images,the datasets contained roughly 80,000 images.

Method	Time (256 px)	Time (512 px)
Gatys et al. [5]	15.863	50.804
WCT [13]	0.689	0.997
Avatar-Net [20]	0.248	0.356
AdaIN [7]	0.011	0.039
ours (`Relu_4_1`)	0.012	0.042
ours (multi-level)	0.017	0.055

Table 1: Execution time comparison (in seconds).

The Authors of the Paper have proposed a new algorithm to perform Style transfer tasks effectively and efficiently by using their proposed SANet architecture in the form of an encoder-decoder module and has achieved better results and quicker runtimes than the existing methods except the AdaIN method which has shown faster run-times.

2.11 LITERATURE SURVEY TABULATED

Paper Name	***Proposed Year***	***Objective/Outcome***
A Universal Style Transfer via Feature Transforms	2017	The study demonstrates a model to transfer styles to content images and produce high quality output. Feed-forward based methods are able to blend content and style images but they are unable to work on unseen styles or low visual quality.
Image Style Transfer Using Convolutional Neural Networks	2016	This study demonstrates an algorithm that is able to separate and recombine the style and content images. It generates new images of high visual quality that blends the content of an input image with the style of various popular paintings.
Neural Style Transfer: A Review	2018	This study presents an overview of Neural Style Transfer Models. It evaluates and compares various algorithms used for NST. Finally it discusses about the several applications and scope for future development.
Deep Photo Style Transfer	2017	The authors of this research paper points the transfer learning concept in a new direction by applying transformations from the input to the output to be locally affine in colorspace of the image, and to express this constraint as a custom fully differentiable energy term.
Audio Style Transfer	2018	The authors of this research paper propose a flexible framework for the task, which uses a sound texture model to extract texture statistics followed by optimization and the stylized signal generation.
Neural Style Transfer for Audio Spectrograms)	2018	The Authors of this Research Paper aim to present a machine learning technique for style transfer on audio signals analogous to the approach of Gatys et al on images by using a convolutional neural network, and investigates the generation of spectrograms from noise
Signal Estimation from Modified Short-Time Fourier Transform	1984	The authors of this research paper aim to present an algorithm to estimate a analogue signal from its modified Short-term Fourier transform.
Fast Signal Reconstruction From Magnitude Stft Spectrogram Based On Spectrogram Consistency	2010	The Research paper presents the latest advancements on the applications of Short-term Fourier Transform in Signal Reconstruction from an altered Spectrogram.

Adam: A Method For Stochastic Optimization	2015	The Authors of the Research paper introduced a Stochastic Gradient optimization algorithm that is computationally efficient and has minimum memory requirement and which can be used for problems which are large in terms of parameters and data and that can be applied to methods with noise in the data or has sparse gradients.
Arbitrary Style Transfer with Style-Attentional Networks	2019	The Research Paper introduces and proposes the use of a novel style attention network called the (SANet) and a new Identity loss function and multi-level feature embeddings to tackle the issue of maintaining local and global style patterns and the content structure.

2.11.1 OBSERVATIONS

The research papers discussed have provided us with information and the statistical performance data of different approaches and machine learning models which aided us in gaining inspiration and the necessary knowledge to successfully find an innovative approach to transfer learning on audio signals.

2.11.2 CONCLUSION

We have studied and interpreted the research papers and converged on an approach to implement Transfer Learning on an audio signal by stitching together different concepts implemented in the papers.

Our Approach is set to employ the preprocessing techniques, and the development of human auditory system inspired CNN and the pipeline architecture in the aim to achieve an enhanced performance and a viable contribution to the field of Transfer Learning

CHAPTER 3

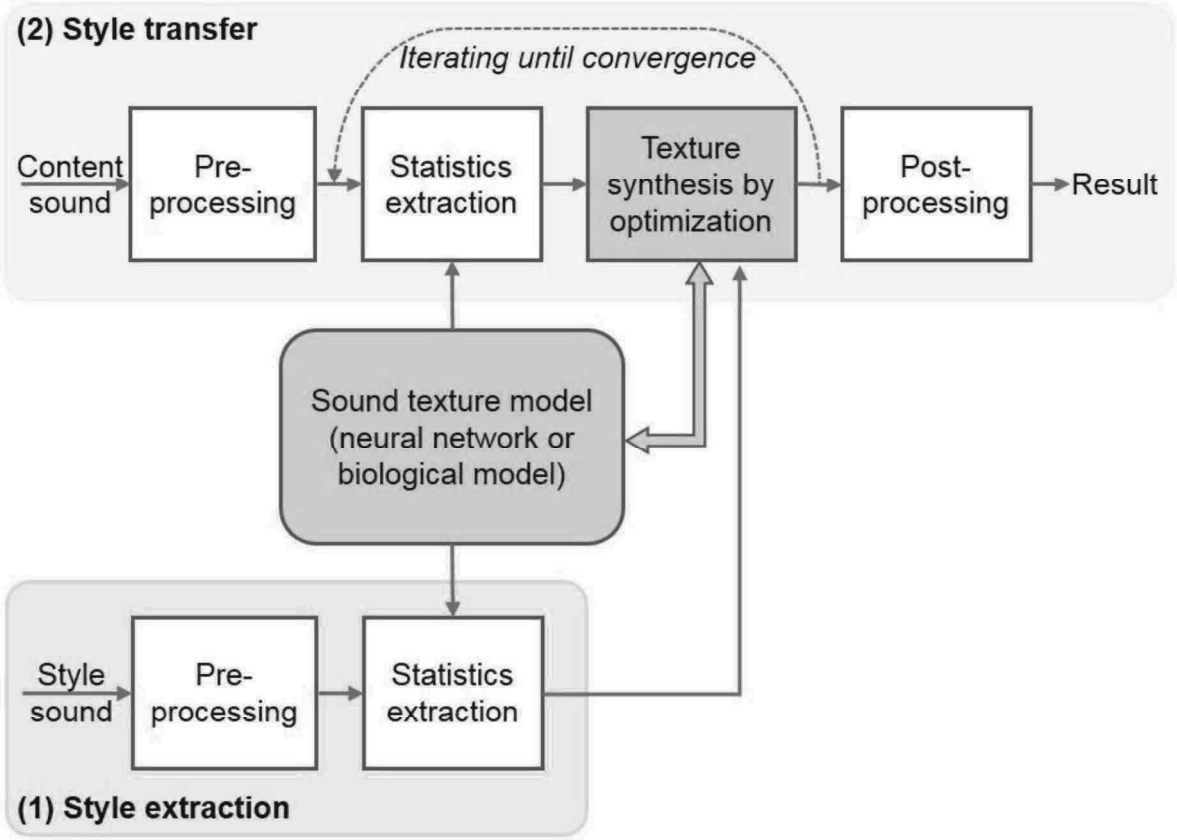
PROPOSED MODEL

3.1 MODEL DESCRIPTION

Our Proposed Model comprises preprocessing, model building and postprocessing. We built our model on tensor flow using Jupyter notebook.
For preprocessing, we used Short Term Fourier transform(STFT) to build a spectrogram from audio input. Spectrogram is a pictorial representation of a one dimensional audio signal.
We had two choices in model building. We could have used a pretrained network like VGGnet or a network with random filters. Present Systems mostly work with pretrained networks but we decided to try a different course. According to our research, shallow random networks give more promising results hence we decided to use CNN with random filters. In statistics extraction, we computed content and style feature maps from our CNN model . Our aim here is to perform texture synthesis that is to produce a feature map with the perfect blend of both content and style feature maps. An optimization algorithm was used to generate it.
Finally postprocessing was performed to convert the blended result feature map to an audio file. Here, we used Griffin Lim algorithm to reconstruct audio.
Our process works on “content” audio and “style” audio independently. We used Short Term Fourier Transform, CNN with random weights and Griffin Lim algorithm to get our desired output audios.
The goal is to achieve the perfect balance of content and style. We have to preserve the content while stylizing it. The proposed model succeeded in creating new music.

3.2 MODEL ARCHITECTURE

Figure 3.2.1 Audio Style Transfer Learning Architecture



1) Preprocessing- This module converts “content” and “style” audio files into spectrograms.

2) Model Building-This module consists Statistics extraction and Texture synthesis. A sound texture model is used to build both content and style feature maps. Then its blended in a desired balance during texture synthesis. Here, an optimization algorithm is used to iteratively optimize the mixed feature map.

3) Postprocessing- This module converts the output spectrogram into audio file.

CHAPTER 4

IMPLEMENTATION

4.1 DATASET

Our proposed model for audio style transfer employs one-dimensional convolutional neural network architecture. The implementation of this model is done under three segments: Preprocessing, model building, post processing.

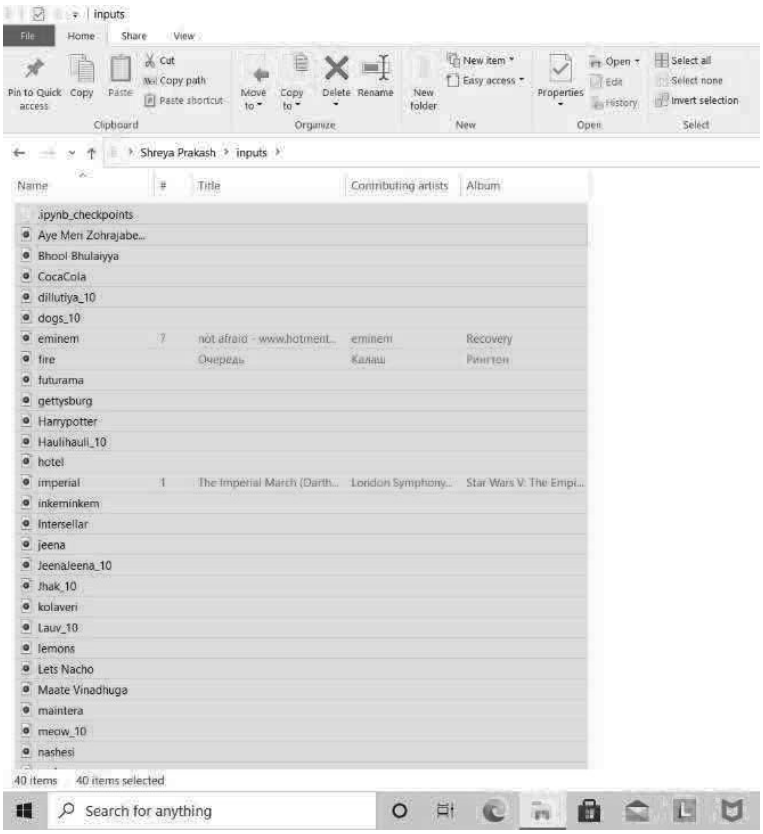
Our model is a one layered convolutional network and it does not require training. We used completely random weights instead of pretrained network as the former gives better results according to our research.

The data used on our model is a mixture of Bollywood, Hollywood and Tollywood songs. We have a **combined 40 total audio files** that we used on this Model.

Its not necessary that any two content and style audios chosen will give pleasant music. Instead, they produced more noise than music. We had to experiment on all the combinations possible from these audio files to generate proper musical tunes.

This dataset is in the form of a .mp3 extension. All the content and style audios are in the input file. All our result audios are in the output file.

FIGURE 4.1.1 AUDIO DATASET (INPUT)



4.2 MODULES

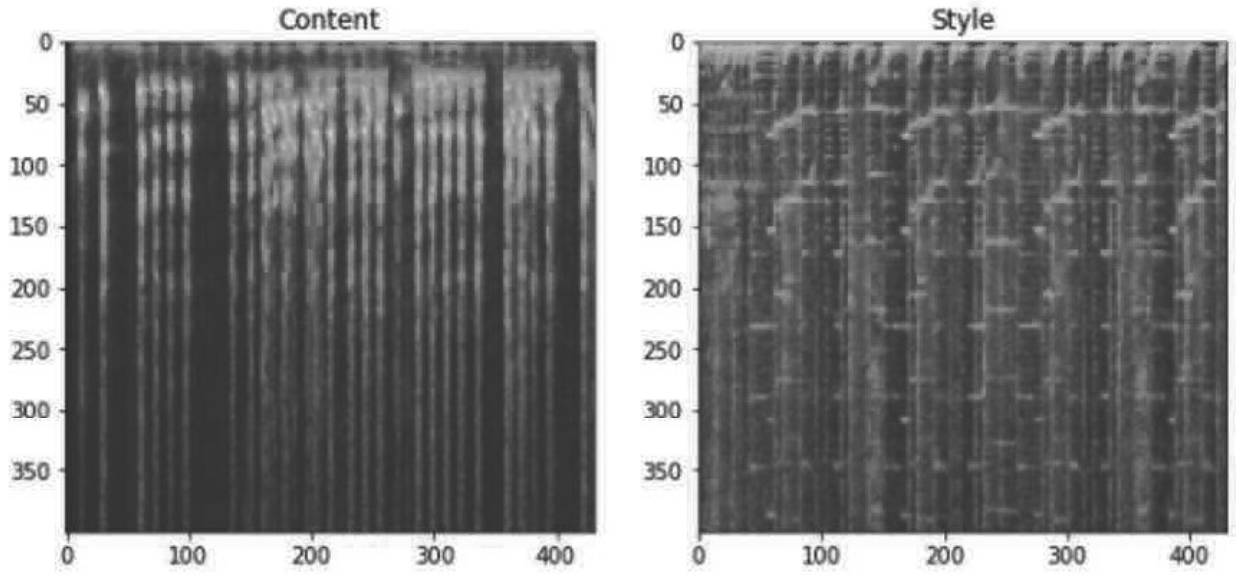
- Tensorflow: This is the platform used for performing the implementation for our module. It is an open source platform for building deep networks
- Librosa: a python package for audio and music analysis. This module is used on our raw dataset to obtain trainable features for the model.
- Matplotlib: python library used for graphical interpretation of data and plotting different types of graphs.
- Numpy: This brings computational powers of programming languages like C and FORTAN to python. This basically facilitates in data handling and representation of dataset in tabular format known as dataframes.
- IPython.display: It allows us to play and stop the audio files

4.3 PREPROCESSING

Preprocessing was done on both content and style audios using Short Term Fourier Transform(STFT). In this module, we imported all the important libraries and the goal was to convert content and style audios into spectrograms.
Spectrogram is a pictorial representation of a one-dimensional audio signal.

Content and Style audios are chosen. We created a function to read the audio files and convert them into spectrograms. Here, Short Term Fourier Transform function was used from the librosa library. This function is performed on both audios. Dimensions of style spectrogram is made equal to that of content.
Both spectrograms are plotted using the matplotlib library.

FIGURE 4.3.1: Spectrograms plotted for content and style audio files from our proposed model



4.4 MODEL BUILDING

We created a CNN model with one convolutional layer and a rectified linear units layer(ReLU) with 4096 filters.

Kernel is a group of random weights which slides across the input matrix to produce a feature map. The rectified linear units layer will output the input directly if it is positive, or else it will output 0.

Content and Style spectrograms are fed to this model to compute Content and Style feature maps. These feature maps are used to generate a loss function between content and style spectrograms and the output spectrogram.

For texture synthesis, we computed style loss and content loss from the generated feature maps. The total sum of content and style loss is fed into the optimizer.
The main optimization algorithm for the style transfer algorithm is 'L-BFGS-B'. Our model iterates till it produces the desired blend of style and content feature maps.

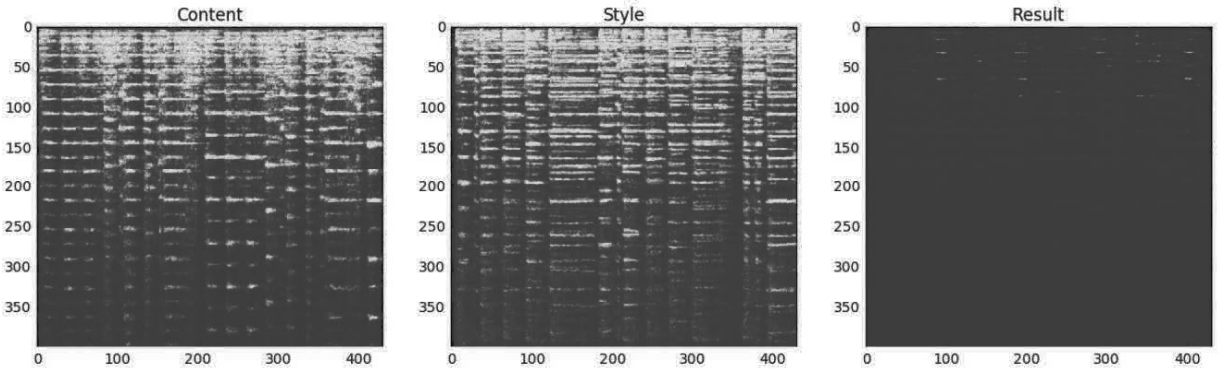
We had to decide the correct balance of content and style in the output. The main parameter in our model which decides this major issue is the “ALPHA” parameter.

Higher the “ALPHA” value more the content in the generated output. Therefore, ALPHA is zero means there is no content in the audio. We had to experiment with different values to get the best output audio results.

4.5 POSTPROCESSING

For Postprocessing, the output feature map has to be converted into an audio file. We used Griffin-Lim algorithm for this purpose. Inverse Short Term Fourier Transform function is performed using the librosa library.
Output audio is generated and saved. Finally we plotted the spectrograms for content, style and result audios to compare.

FIGURE 4.5.1: Spectrograms plotted for content, style and result audio signals.



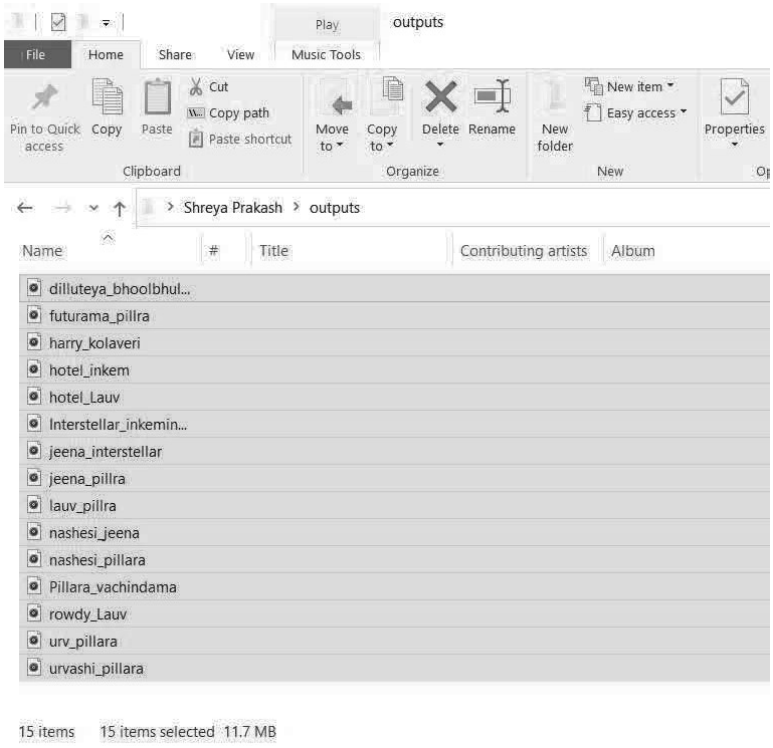
Here is an example of one of our result audios. The lines are less bright than content and style spectrograms. This shows the amplitude of the signal. If we look closely, we can see few areas with moderate brightness.

CHAPTER 5

RESULTS

We experimented on many different kinds of input music audios and were able to create new fresh tunes.
These output audio files are created from our audio style transfer learning model using different combinations of input audio files for “style” and “content”.

Figure 5.1: Output Audio files



CHAPTER 6

CONCLUSION

Our project can be further developed for experimenting on speech audio files.

The variables used in our project is suitable for musical tunes not speech. Hence, when experimented with speech audios, the output generated was mostly noise. When we used guitar and piano tunes for content and style audios, the generated output was pleasant.

A better understanding of music is required to decide what “content” and “style” audios should be like. This understanding can be used to find representations of audio other than Short Term Fourier Transform, a better model architecture and appropriate content and style audio files to use on the model.

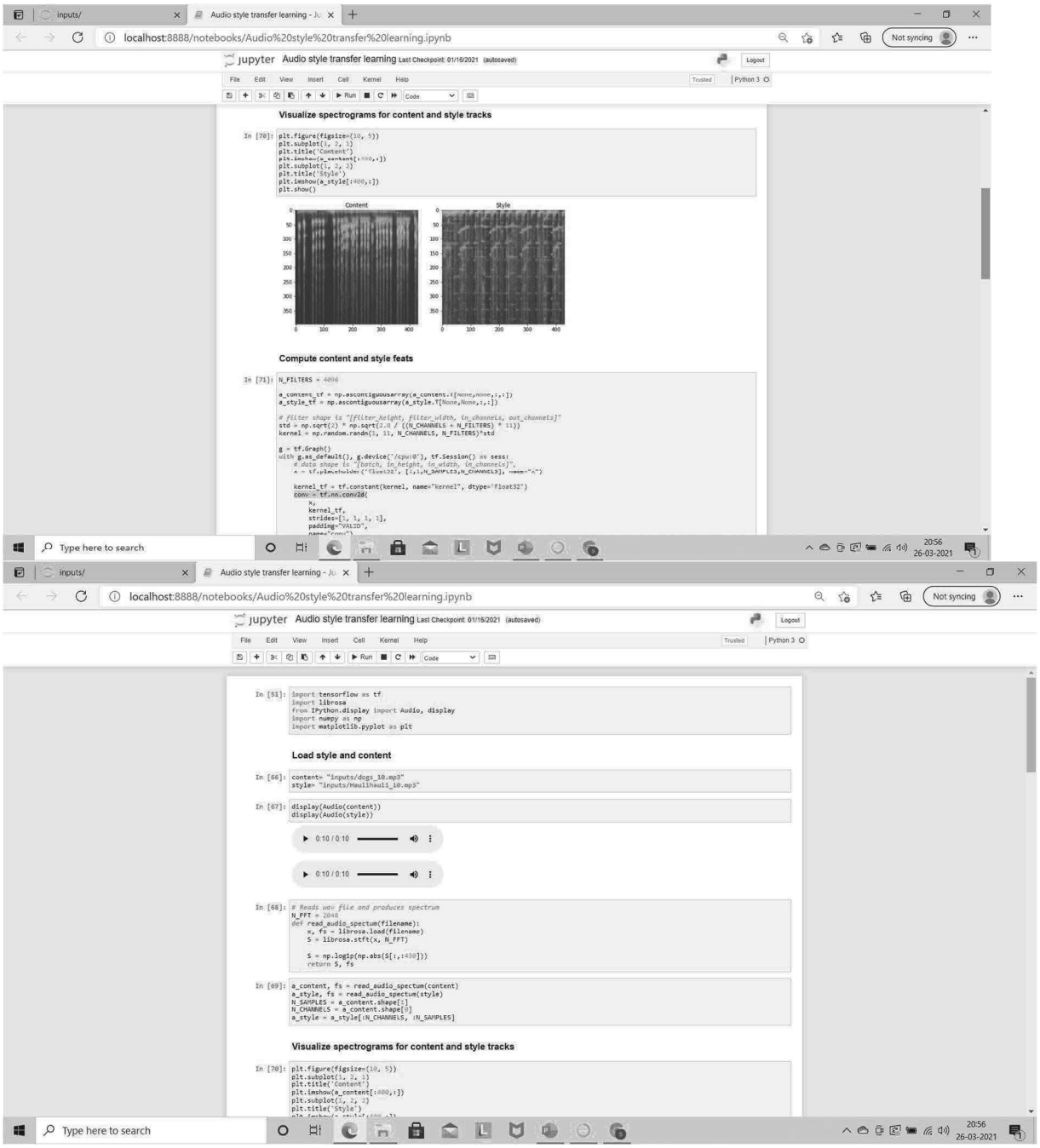
REFERENCES

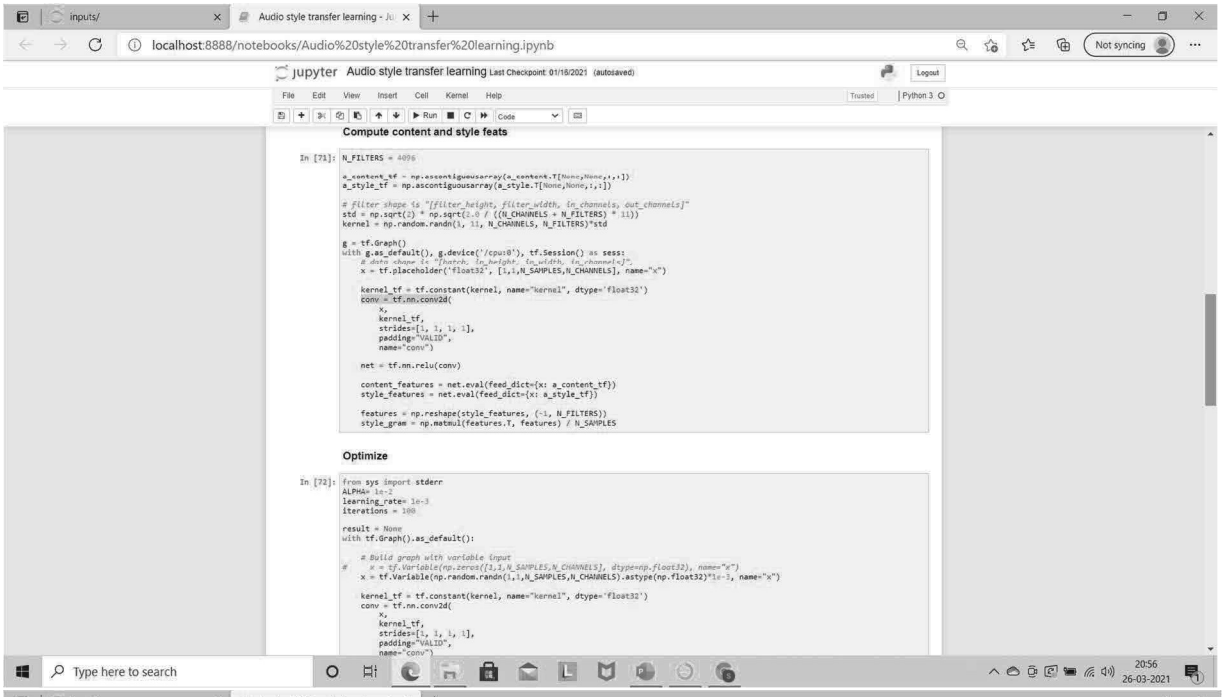
1. Eric Grinstein, Ngoc Q. K. Duong Alexey Ozerov, Patrick P´erez, “Audio style transfer,” presented in IEEE International Conference on Acoustics, Speech and Signal Processing, 2018.
2. Prateek Verma, Julius O. Smith " Neural Style Transfer for Audio Spectrograms" Appeared in 31st Conference on Neural Information Processing Systems, 2018.
3. Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, Ming-Hsuan Yang, " Universal Style Transfer via Feature Transforms " Accepted by NIPS 2017.
4. Leon A. Gatys, Alexander S. Ecker, Matthias Bethge " Image Style Transfer Using Convolutional Neural Networks" presented in IEEE Conference on Computer Vision and Pattern Recognition, 2016.
5. Yongcheng Jing, YezhouYang, Zunlei Feng, JingwenYe, Mingli Song, " Neural Style Transfer: A Review" presented in 2018.
6. Fujun Luan, Sylvain Paris, Eli Shechtman, Kavita Bala, " Deep Photo Style Transfer" presented in IEEE Conference on Computer Vision and Pattern Recognition, 2017.
7. Daniel W. Griffin, Jae S. Lim" Signal Estimation from Modified Short-Time Fourier Transform" **Published in** IEEE Transactions on Acoustics, Speech, and Signal Processing, Apr 1984
8. Jonathan Le Roux, Hirokazu Kameoka, Nobutaka Ono, Shigeki Sagayama "Fast Signal Reconstruction From Magnitude Stft Spectrogram Based On Spectrogram Consistency" Proc. of International Conference on Digital Audio Effects, 2010.
9. Diederik P. Kingma, Jimmy Lei Ba "Adam: A Method For Stochastic Optimization" Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015
10. Dae Young Park, Kwang Hee Lee “Arbitrary Style Transfer with Style-Attentional Networks”; presented in IEEE Conference on Computer Vision and Pattern Recognition, 2017.

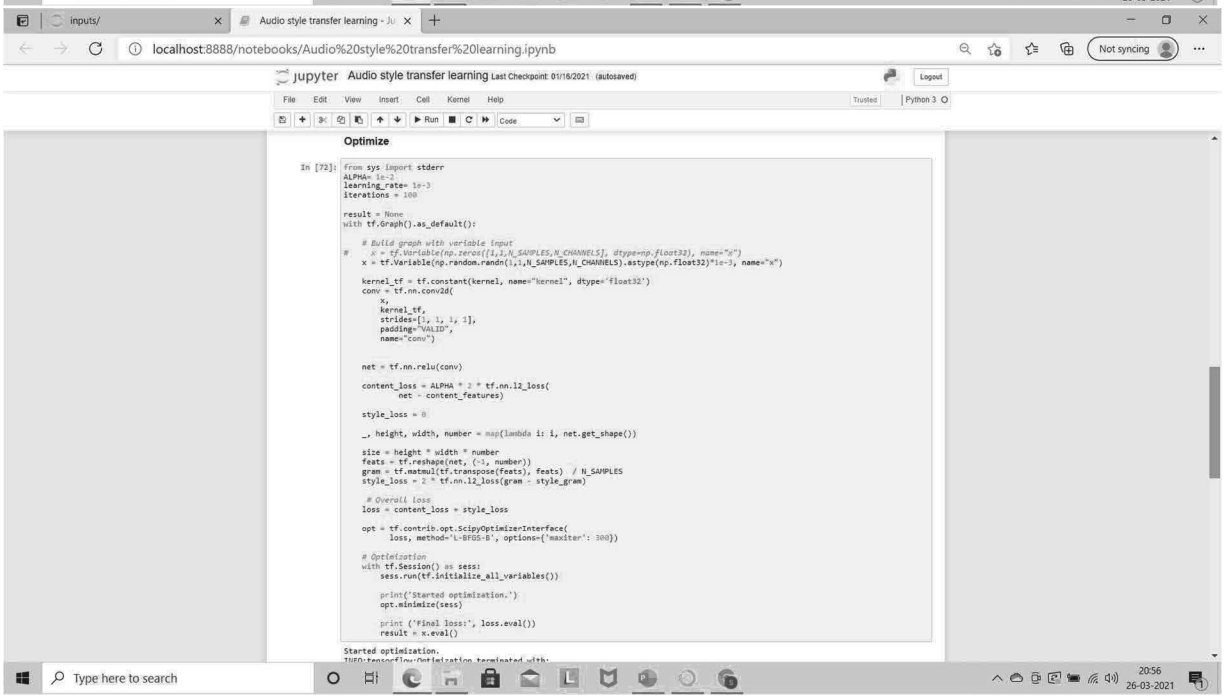
APPENDIX A

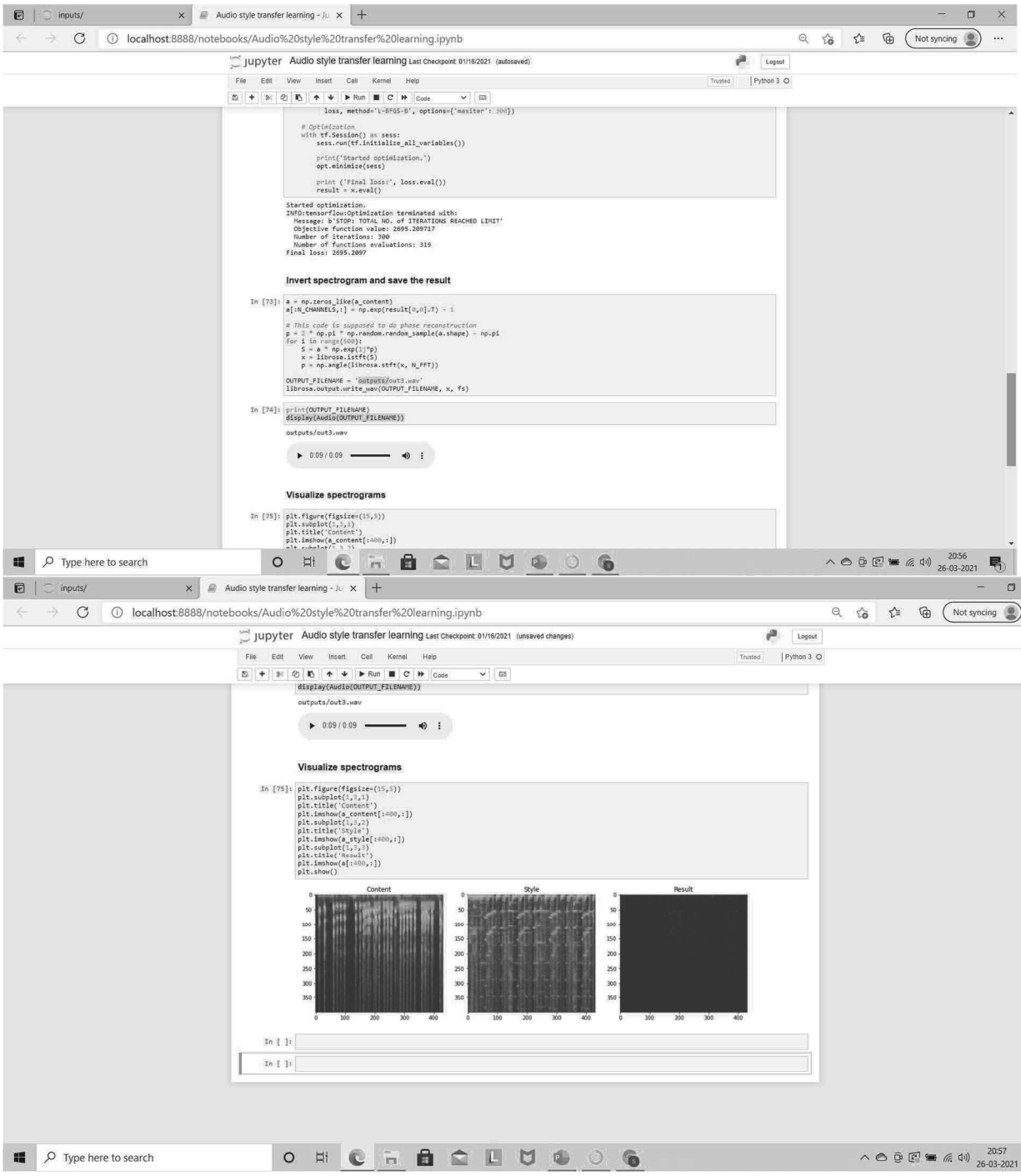
CODE

A.1 AUDIO STYLE TRANSFER LEARNING MODEL




inputs/
Audio style transfer learning - Ju
localhost:8888/notebooks/Audio%20style%20transfer%20learning.ipynb
Not syncing
jupyter Audio style transfer learning Last Checkpoint: 01/16/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
Compute content and style feats
In [71]:
N_FILTERS = 4096
a_content_tf = np.ascontiguousarray(a_content.T[None,None,:,:])
a_style_tf = np.ascontiguousarray(a_style.T[None,None,:,:])
# filter shape is "[filter_height, filter_width, in_channels, out_channels]"
std = np.sqrt(2) * np.sqrt(2.0 / ((N_CHANNELS + N_FILTERS) * 11))
kernel = np.random.randn(1, 11, N_CHANNELS, N_FILTERS)*std
g = tf.Graph()
with g.as_default(), g.device('/cpu:0'), tf.Session() as sess:
# data shape is "[batch, in_height, in_width, in_channels]",
x = tf.placeholder('float32', [1,1,N_SAMPLES,N_CHANNELS], name="x")
kernel_tf = tf.constant(kernel, name="kernel", dtype='float32')
conv = tf.nn.conv2d(
x,
kernel_tf,
strides=[1, 1, 1, 1],
padding="VALID",
name="conv")
net = tf.nn.relu(conv)
content_features = net.eval(feed_dict={x: a_content_tf})
style_features = net.eval(feed_dict={x: a_style_tf})
features = np.reshape(style_features, (-1, N_FILTERS))
style_gram = np.matmul(features.T, features) / N_SAMPLES
Optimize
In [72]:
from sys import stderr
ALPHA= 1e-2
learning_rate= 1e-3
iterations = 100
result = None
with tf.Graph().as_default():
# Build graph with variable input
# x = tf.Variable(np.zeros([1,1,N_SAMPLES,N_CHANNELS], dtype=np.float32), name="x")
x = tf.Variable(np.random.randn(1,1,N_SAMPLES,N_CHANNELS).astype(np.float32)*1e-3, name="x")
kernel_tf = tf.constant(kernel, name="kernel", dtype='float32')
conv = tf.nn.conv2d(
x,
kernel_tf,
strides=[1, 1, 1, 1],
padding="VALID",
name="conv")
Type here to search
20:56
26-03-2021


inputs/
Audio style transfer learning - Ju
localhost:8888/notebooks/Audio%20style%20transfer%20learning.ipynb
Not syncing
jupyter Audio style transfer learning Last Checkpoint: 01/16/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
Optimize
In [72]:
from sys import stderr
ALPHA= 1e-2
learning_rate= 1e-3
iterations = 100
result = None
with tf.Graph().as_default():
# Build graph with variable input
# x = tf.Variable(np.zeros([1,1,N_SAMPLES,N_CHANNELS], dtype=np.float32), name="x")
x = tf.Variable(np.random.randn(1,1,N_SAMPLES,N_CHANNELS).astype(np.float32)*1e-3, name="x")
kernel_tf = tf.constant(kernel, name="kernel", dtype='float32')
conv = tf.nn.conv2d(
x,
kernel_tf,
strides=[1, 1, 1, 1],
padding="VALID",
name="conv")
net = tf.nn.relu(conv)
content_loss = ALPHA * 2 * tf.nn.l2_loss(
net - content_features)
style_loss = 0
_, height, width, number = map(lambda i: i, net.get_shape())
size = height * width * number
feats = tf.reshape(net, (-1, number))
gram = tf.matmul(tf.transpose(feats), feats) / N_SAMPLES
style_loss = 2 * tf.nn.l2_loss(gram - style_gram)
# Overall loss
loss = content_loss + style_loss
opt = tf.contrib.opt.ScipyOptimizerInterface(
loss, method='L-BFGS-B', options={'maxiter': 300})
Optimization
with tf.Session() as sess:
sess.run(tf.initialize_all_variables())
print('Started optimization.')
opt.minimize(sess)
print ('Final loss:', loss.eval())
result = x.eval()
Started optimization.
Type here to search
20:56
26-03-2021


inputs/
Audio style transfer learning - Ju
localhost:8888/notebooks/Audio%20style%20transfer%20learning.ipynb
Not syncing
jupyter Audio style transfer learning Last Checkpoint: 01/16/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
loss, method='L-BFGS-B', options={'maxiter': 300})
Optimization
with tf.Session() as sess:
sess.run(tf.initialize_all_variables())
print('Started optimization.')
opt.minimize(sess)
print ('Final loss:', loss.eval())
result = x.eval()
Started optimization.
INFO:tensorflow:Optimization terminated with:
Message: b'STOP: TOTAL NO. of ITERATIONS REACHED LIMIT'
Objective function value: 2695.209717
Number of iterations: 300
Number of functions evaluations: 319
Final loss: 2695.2097
Invert spectrogram and save the result
In [73]:
a = np.zeros_like(a_content)
a[:N_CHANNELS,:] = np.exp(result[0,0].T) - 1
# This code is supposed to do phase reconstruction
p = 2 * np.pi * np.random.random_sample(a.shape) - np.pi
for i in range(500):
S = a * np.exp(1j*p)
x = librosa.istft(S)
p = np.angle(librosa.stft(x, N_FFT))
OUTPUT_FILENAME = 'outputs/out3.wav'
librosa.output.write_wav(OUTPUT_FILENAME, x, fs)
In [74]:
print(OUTPUT_FILENAME)
display(Audio(OUTPUT_FILENAME))
outputs/out3.wav
0:09 / 0:09
Visualize spectrograms
In [75]:
plt.figure(figsize=(15,5))
plt.subplot(1,3,1)
plt.title('Content')
plt.imshow(a_content[:400,:])
Type here to search
20:56
26-03-2021
inputs/
Audio style transfer learning - Ju
localhost:8888/notebooks/Audio%20style%20transfer%20learning.ipynb
Not syncing
jupyter Audio style transfer learning Last Checkpoint: 01/16/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
display(Audio(OUTPUT_FILENAME))
outputs/out3.wav
0:09 / 0:09
Visualize spectrograms
In [75]:
plt.figure(figsize=(15,5))
plt.subplot(1,3,1)
plt.title('Content')
plt.imshow(a_content[:400,:])
plt.subplot(1,3,2)
plt.title('Style')
plt.imshow(a_style[:400,:])
plt.subplot(1,3,3)
plt.title('Result')
plt.imshow(a[:400,:])
plt.show()
Content
Style
Result
In [ ]:
In [ ]:
Type here to search
20:57
26-03-2021