



Audio Style Transfer

Brennan Mahoney, Roger Finnerty, Jack Edelist, Amruth Niranjan,
Zane Mroue

Boston University CS/ECE 523 Deep Learning Project

Motivation & Task

Initial motivation: change the gender/accent of song acapellas

Potential challenges:

- Computationally expensive training procedures
- Difficult to capture nuances between accents
- Audio reconstruction using spectrograms

Approach:

- Perform style transfer on shorter speech samples: gender, instrument, accent
- Compare 4 models for style transfer of Mel Spectrograms
 - 1-layer CNN, VGG-19, Vision Transformer, ResNet
- Implement a DL model for audio reconstruction
- Qualitative comparison of style-transferred audio

Audio vs. Image Data

Spectrograms:

- More memory-intensive due to 2-D nature
- Adds processing time to convert spectrograms to audio

Raw audio:

- No need for STFT and inverse operations: higher computational efficiency
- BUT, lacks visual amplitude, phase, frequency over time data that NNs can learn from

Related work

[1] "Image style transfer using convolutional neural networks."

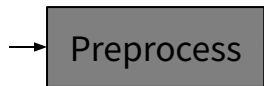
- Utilizes a pre-trained CNN to separate and recombine content and style from distinct images.
- Employs a loss function that minimizes the difference in content and style between the target and generated images.
- Demonstrates successful style transfer by adjusting the feature correlations in convolutional layers.

[2] "How to Convert Audio to Mel-Spectrogram to Audio."

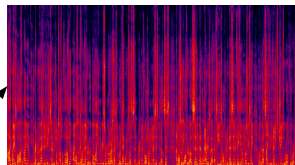
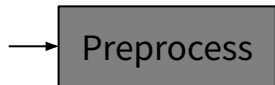
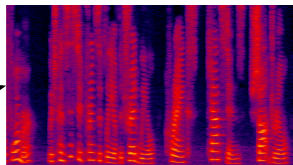
- Describes the conversion of audio to mel-spectrograms using the Librosa library, focusing on feature extraction for audio analysis.
- Details the inverse process to reconstruct audio from mel-spectrograms, emphasizing preservation of audio characteristics.
- Provides Python code snippets to facilitate hands-on application of theoretical concepts in audio signal processing.

Approach

Step 1:
Obtain
.wav data



Step 2:
Convert the
audio files to
spectrogram
images

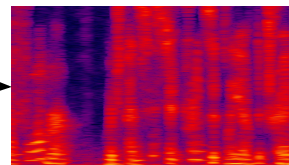


Step 3:
Execute style
transfer

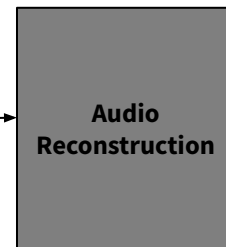


VGG19
ResNet
Vision Transformer

Step 4: Obtain
output in image
format



Step 5:
Convert the
spectrogram
back to .wav
data



Datasets

- ImageNet (VGG and ViT pretraining)
 - 14M + images, standard CV benchmark dataset
- Content audio: LJ Speech Dataset
 - Public domain speech dataset (from audiobooks)
 - 13,100 short audio clips (1 - 10 sec)
- Style audio: variety of clips (gender, accents, instruments)
 - English Multi-speaker Corpus for CSTR Voice Cloning Toolkit

[Audio → Mel Spectrogram → Audio] Conversion

Mel Spectrograms:

- Applies a frequency-domain filter bank to audio signals that are windowed in time
 - Uses the Mel Scale on y-axis (non-linear transformation of the frequency scale)
1. Short-time Fourier Transform + Griffin-Lim algorithm
 - a. STFT: visual representation of the STFT that displays the frequencies present in an audio signal as they change over time
 - b. GLA: phase reconstruction method that estimates signals from modified short-time Fourier transforms
 2. WaveNet
 - Deep neural network for generating raw audio

Initial Results: Single-Layer CNN [STFT & GLA] (1/2)

num_epochs = 20000

learning_rate = 0.002

style_param = 1

content_param = 1e2

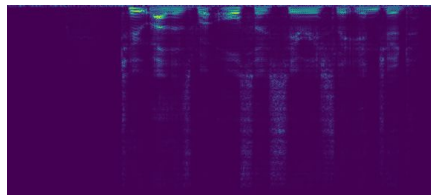
```
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

        # 2-D CNN
        self.conv1 = nn.Conv2d(1, OUT_CHANNELS, kernel_size=(3, 1), stride=1, padding=0)
        self.LeakyReLU = nn.LeakyReLU(0.2)

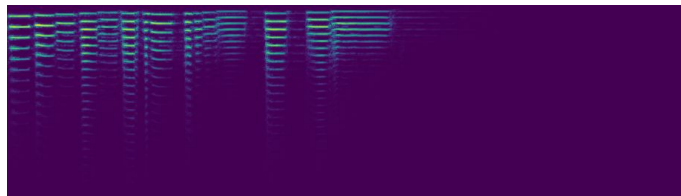
        # Set the random parameters to be constant
        weight = torch.randn(self.conv1.weight.data.shape)
        self.conv1.weight = torch.nn.Parameter(weight, requires_grad=False)
        bias = torch.zeros(self.conv1.bias.data.shape)
        self.conv1.bias = torch.nn.Parameter(bias, requires_grad=False)

    def forward(self, x_delta):
        out = self.LeakyReLU(self.conv1(x_delta))
        return out
```

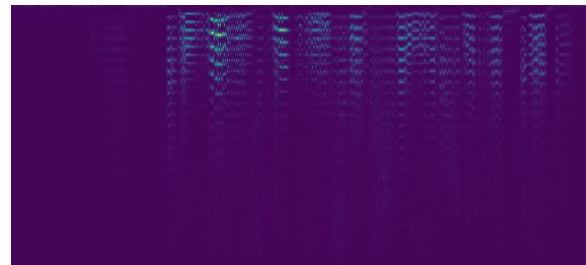

Initial Results: Single-Layer CNN (2/2)



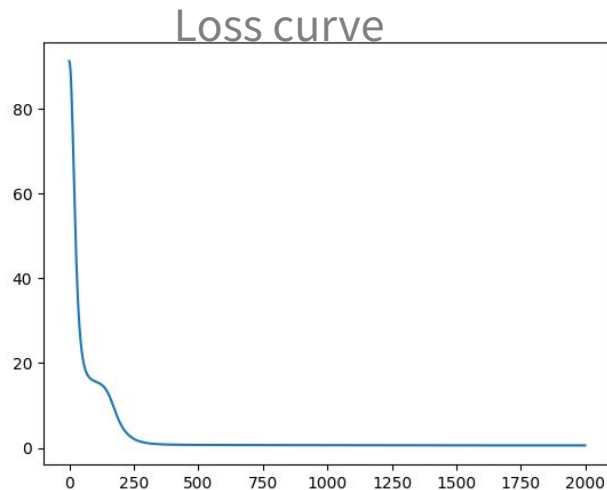
Content spec



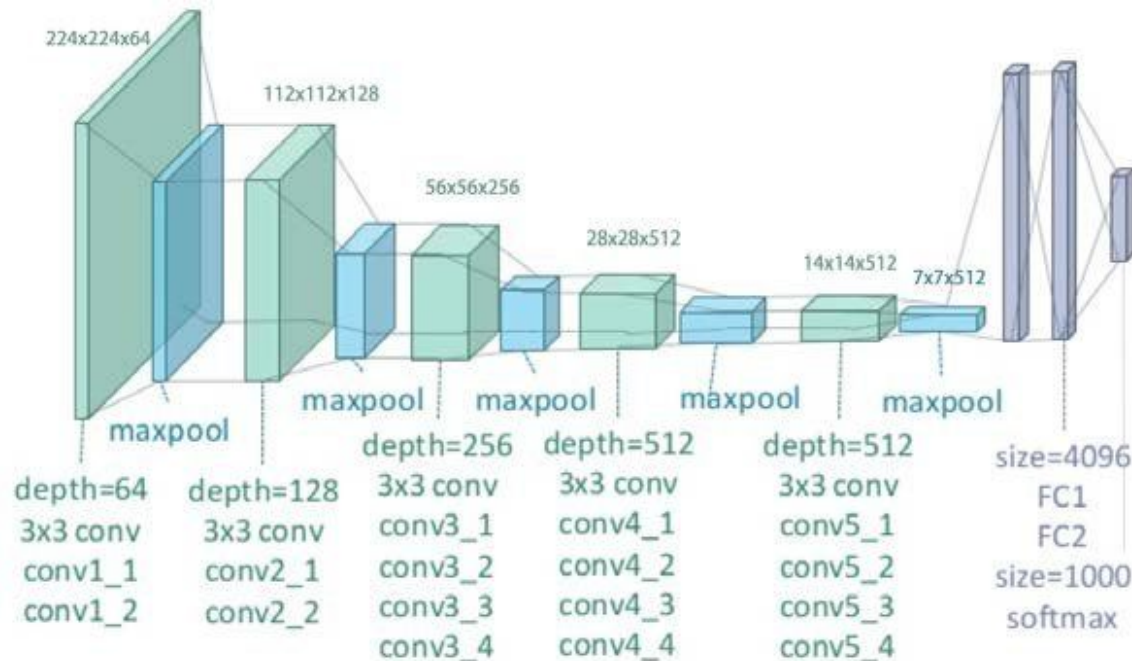
Style spec



Generated spec



Style Transfer: VGG19 Backbone (Gatys et al)



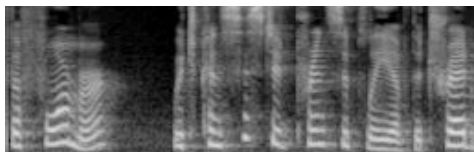
$$\mathcal{L}_{content}(p, x, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

$$\mathcal{L}_{style}(a, x) = \sum_{l=1}^L w_l E_l$$

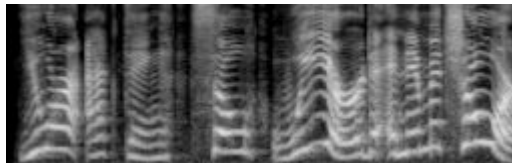
$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

Results for VGG19 (Accent transfer)

Content Spectrogram



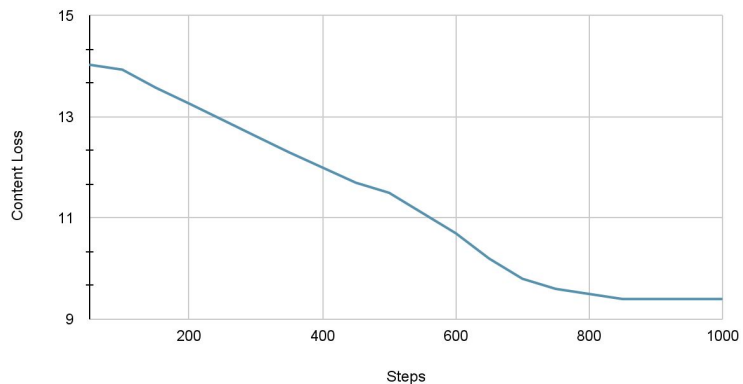
Style Spectrogram



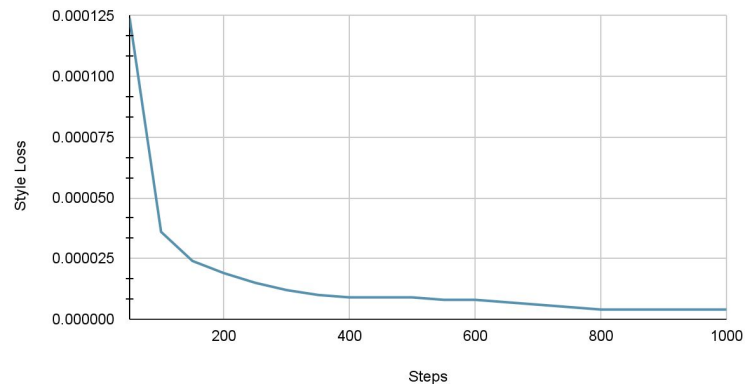
Output Spectrogram



Content Loss vs Steps

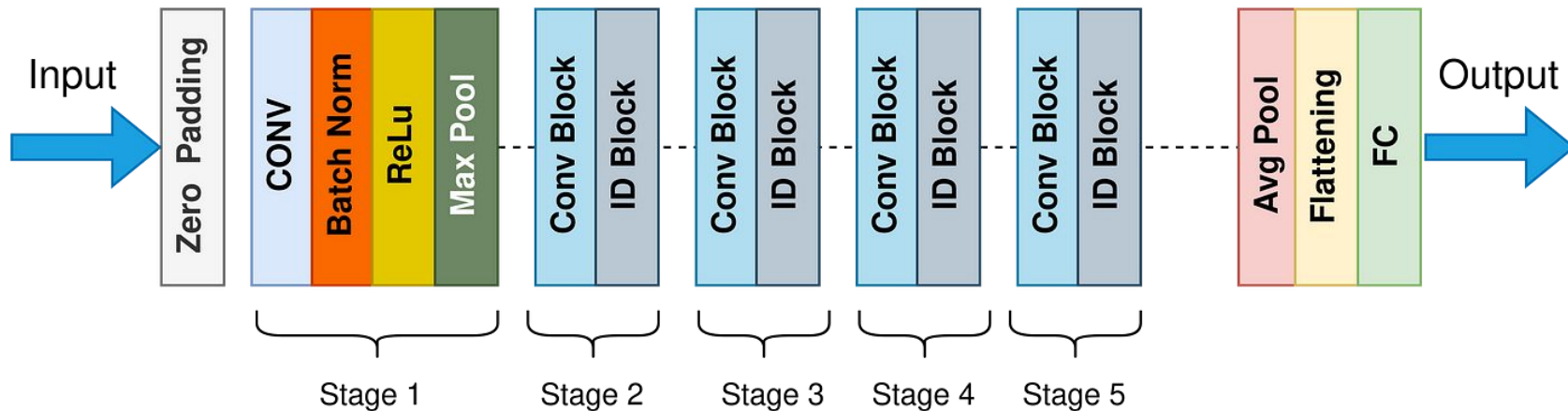


Style Loss vs Steps



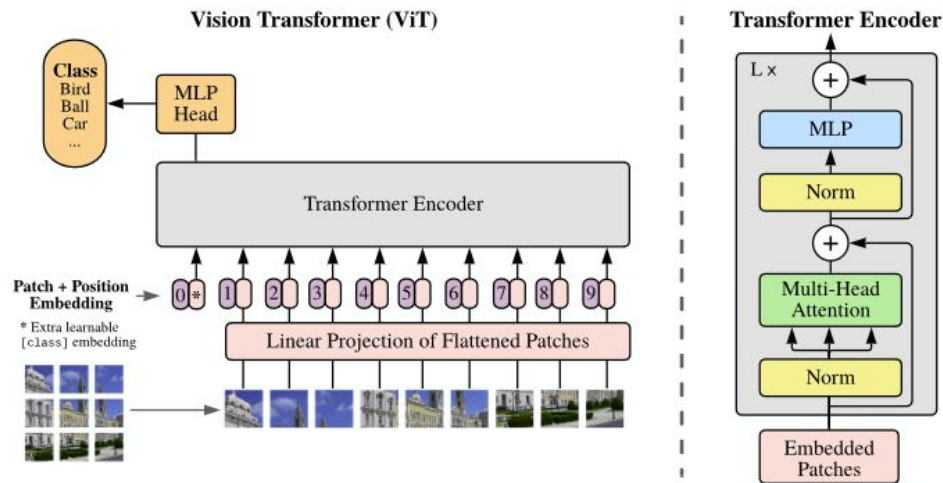
Style Transfer: ResNet Backbone

ResNet50 Model Architecture



- ResNets are built out of blocks called "residual blocks" or "residual units." Each block has a shortcut or "skip connection" that allows the input to the block to be added directly to the output of the block. Deeper layers perform at least as well as the shallower ones
- The residual learning framework allows deeper ResNet models to perform complex transformations without losing low-level details, which is crucial for capturing and modifying nuanced audio characteristics like accents or speech styles

Style Transfer: Vision Transformer



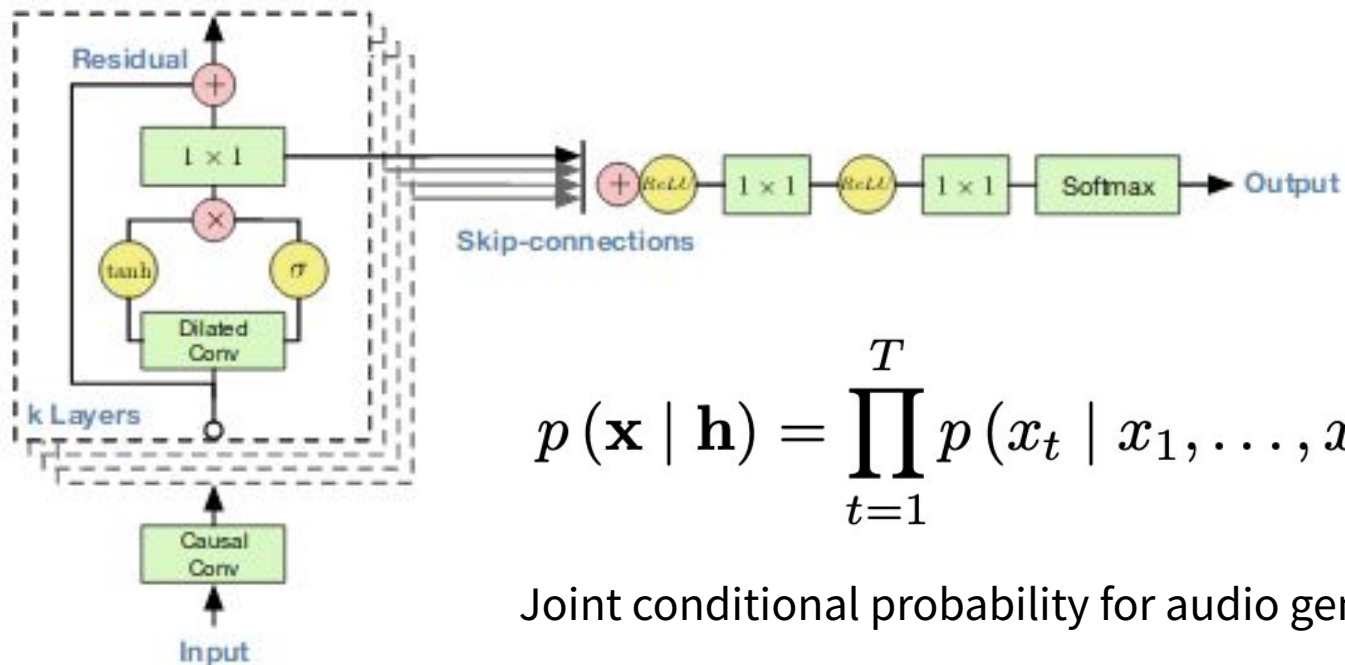
```
content_losses = []
style_losses = []
content_features = img_model(content_img).detach()
style_features = img_model(style_img).detach()
for name, layer in vision_transformer.named_children():
    if name == 'transformer':
        for module_name, module_layer in layer.named_children():
            if module_name == 'blocks':
                for block_name, block_layer in module_layer.named_children():
                    if isinstance(block_layer, V.transformer.Block):
                        i += 1
                        newname = 'Block_{}'.format(i)
                    else:
                        raise RuntimeError('Unrecognized layer: {}'.format(layer.__class__.__name__))

img_model.add_module(newname, block_layer)
# print(newname)
if newname in content_layers:
    target = content_features
    content_loss = ContentLoss(target)
    img_model.add_module("content_loss_{}".format(i), content_loss)
    content_losses.append(content_loss)
if newname in style_layers:
    target_feature = style_features
    style_loss = StyleLoss(target_feature)
    img_model.add_module("style_loss_{}".format(i), style_loss)
    style_losses.append(style_loss)
```

- Model used contains 12 embedding blocks (middle)
- NST feature extraction executed at the output of each block

WaveNet Architecture

Probabilistic model for generating raw audio











$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{t=1}^T p(x_t \mid x_1, \dots, x_{t-1}, \mathbf{h}).$$









Joint conditional probability for audio generation

Experimental Results

1 Layer CNN + STFT + Griffin-Lim Algorithm









Content Audio	Style Audio	Output from style transfer
	Gender 	
	Instrument 	
	Accent 	

VGG19 backbone + WaveNet






Content Audio	Style Audio	Output from style transfer
	Gender 	
	Instrument 	
	Accent 	

Experimental Results (cont.)

ResNet backbone + WaveNet

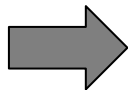
Content Audio	Style Audio	Output from style transfer
	Gender 	
	Instrument 	
	Accent 	

(Future work): VGG19 + ESRGAN (super resolution) + WaveNet

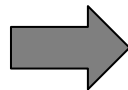
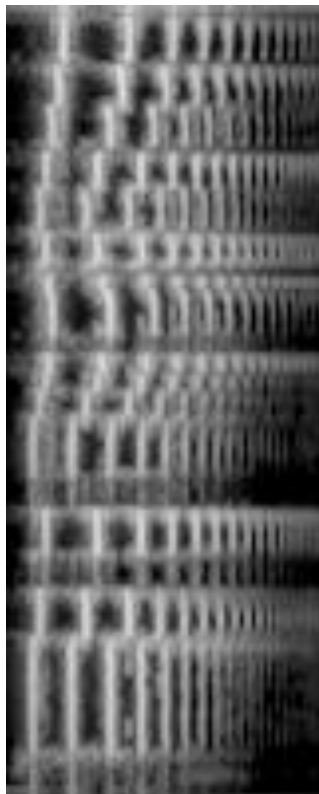
Content Audio	Style Audio	Output from style transfer
	Gender 	
	Instrument 	
	Accent 	

Super Resolution (ESRGAN)

Content



Style



Conclusion + Future Work

- Spectrograms alone fail to capture subtle differences between accents (for speakers of the same gender)
- Recurrent architectures (ViT) struggle to encode any meaningful information at all (at the moment)
- Training should be targeted to **one** specific style transfer task (accents, timbre, pitch, gender, etc.)
- Future work:
 - Super resolution to increase spatial dimension of spectrograms
 - Develop new methods to perform NST using recurrent architectures like ViT
 - Supervised learning → add a label to audio clip (ground truth accent), like conditional diffusion

References

- [1] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. [URL](#)
- [2] Gaurav, "How to Convert Audio to Mel-Spectrogram to Audio." *Kaggle*, www.kaggle.com/code/gaurav41/how-to-convert-audio-to-mel-spectrogram-to-audio. Accessed 29 Apr. 2024.
- [3] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).

Thank you!

Questions?

