Audio-StyleCLIP

Sreevatsank Kadaveru
University of California San Diego
skadaveru@ucsd.edu

Preyaa Patel
University of California San Diego
pspatel@ucsd.edu

Varun Pawar
University of California San Diego
vpawar@ucsd.edu

*Note: all are co-authors

1 Introduction

Image style transfer is a fascinating technique that is able to morph the "artistic style" of a target image without altering its semantic content. It is able to translate the semantic content of the target image from its original artistic style to that of the artistic style of an input style image. An example of image style transfer is illustrated in Figure 1. The prompt of the styling input is incorporated into the semantics of the target image while maintaining the context of the target image.

Image style transfer is a fascinating subject with broad applications in VR, digital art and digital asset creation, image editing tools, and social media filters and artistic apps. Recent developments in neural network techniques [1] [2] enable fast, accurate, and high-quality digital assert, creating and dramatically improving the efficiency of the digital assert designers'. The speed and accuracy of these neural network techniques have enabled the possibility for real-time applications [3].

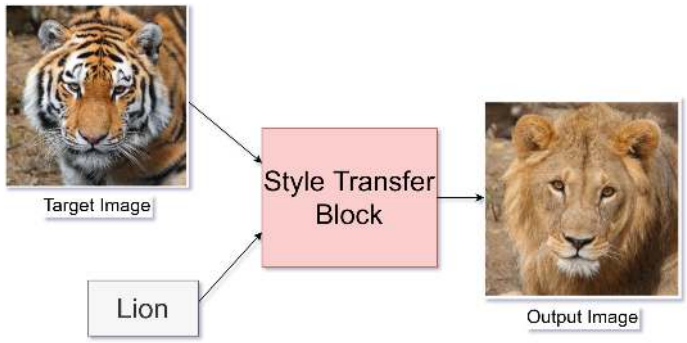


Figure 1: An example of Image Style Transfer

2 Related Work

2.1 Evolution of StyleGAN

Traditionally, image style transfer was achieved using manual modeling and mapping the image textures. These techniques do not generalize to other styles and need to be manually configured for each style, and thereby are inefficient and do not scale. [4] is a first of its kind convolution neural network (CNN) based solution for the image style transfer problem, a seminal paper garnering considerable attention from the computer vision research community and has invigorated interest

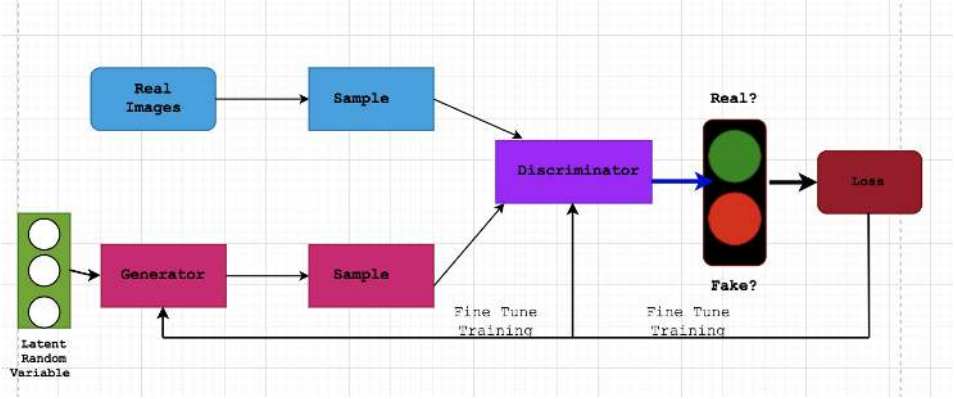


Figure 2: GAN architecture (Source [7])

in the image style transfer problem [1]. This class of techniques is known as Neural Style Transfer (NST). Following this development, several different neural network architectures were explored, of which generative adversarial networks (GAN) have significantly outperformed other classes of techniques [5].

GAN is a two network architecture, which consists of a generative network and an adversarial network, which are pitted against each other in a two-person zero sum game, system architecture illustrated in Figure 2. The generator network is tasked with the generation of realistic fake art while the adversarial network attempts to differentiate between the generated fake art and the real art [5]. The promise of GANs inspired the developments of several variants [1] [2] - Conditional GAN (CGAN), Deep Convolution GAN (DCGAN), StarGAN, and the StyleGAN [6].

2.2 StyleCLIP

The main research paper that we referred to for this project, a text-based StyleGAN, was StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery [8]. In this paper, they use Contrastive Language-Image Pre-training (CLIP) models to make the styling more robust so that it is not confined to the image manipulations that it has seen before. CLIP is an OpenAI contribution [9], which is a neural network trained (image, text) pairs. They combine CLIP with StyleGANs to style images according to the given text using the following three methods.

- In the first method, they do latent optimization using text, where they use CLIP loss instead of ID loss. However, this method is time-consuming.
- In this method, they start with the target image, and the mapper makes a move in the latent $W+$ space.
- The third method involves mapping text instruction to a global direction in the style space of StyleGAN. This gives us the freedom to toggle with the disentanglement intensity.

2.3 AudioCLIP

For audio-driven image styling, we follow the research by [10] in the paper AudioCLIP. AudioCLIP is a trimodal hybrid architecture used to find similarities between the image, text, and speech-based captions pairwise or all three simultaneously. The model extends the existing CLIP modality by adding an audio head to the network. The audio-head is trained using ESResNeXt architecture developed by [11] previously. It uses ResNet-50 with self-attention for generating image heads and transformers-based text-encoders. Traditional CLIP maximizes the cosine similarity between image and text-based representations. It is done by minimizing the symmetric cross-entropy error over the similarity measure. The above loss modality is extended in the AudioCLIP for text-speech and speech-image pairs. This architecture allows the user to simultaneously find image-, audio-, and speech-head and generate corresponding similarity scores.

2.4 FastSpeech

There are no readily-available data sets of human face images with speech prompts. For this reason, we use Coqui TTS [12], an open-source API for text-to-speech(TTS) synthesis. We leverage the API and StyleCLIP dataset for a speech-prompt generation. The API uses FastSpeech Network [13], FastSpeech is an end-to-end text-to-speech(TTS) synthesis network. The network uses a parallel feedforward module for a mel-spectrogram generation, making this network considerably faster than its predecessors. FastSpeech consists of multiple FFT layers with intermediate length regulators. FFT layer consists of multi-head attention followed by 1-D convolution. Its output is given to the length regulator, consisting of a duration predictor block. The block expands the phoneme input so that spectrogram and phoneme sequences are the same lengths. Duration predictor block consists of 1D-convolutional layers with batch normalization and dropout followed by linear layer. Because of feedforward parallelized architecture, the network is 38x faster than its predecessors(as reported by the authors).

3 Motivation

Style transfer is helpful for many reasons and in many sectors in the world. Moreover, with the continued enhancements in AI-driven technology, quick and interactive style transfer can come in very handy to automate manual work. Some application areas that require synthetic image generation like medical imaging, gaming, education, or others require making changes to input images like town planning, architecture, interior designing, advertising, and Virtual Reality.

There are many existing pieces of work wherein they style images using text instructions. Nevertheless, as text-based styling using StyleGANs is relatively new, we are motivated to make this process faster, more convenient, and more interactive by taking speech instructions instead of text to style the input image accordingly.

4 Objective

We aim to develop a novel style-transfer technique that styles a target image based on an input audio description of the style. We take inspiration from recent developments in the domains of natural language processing and style-transfer such as StyleCLIP [8], OpenAI CLIP [9], and StyleGAN [6].

We desire to inherit the zero-shot inference capabilities of the CLIP model and extend the same to the audio domain.

5 System Architecture

The system inputs a target image and a speech signal and outputs a stylized image. It is composed of multiple sub-networks, such as the image-encoder, latent mapper, and image decoder. The image-encoder block projects an input image onto a latent space $\mathcal{W}+$. We use OpenAI's CLIP [9] model for the image encoder, and thereby the projection space of the image is the latent space of the CLIP model.

The latent mapper is a network designed to perform edits to the style of the target image (input image). This is done by constructing a perturbation δw to the $w \in \mathcal{W}+$ produced by the image encoding network in a direction to impart the intended style onto the target image. We denote the latent mapper as a transformation function ($\mathcal{P}_l$) in the $\mathcal{W}+$ space (see Eq 1). The perturbation δw is applied to the latent embedding as shown in Eq 3 to obtain the augmented $w^{'}$ vector.

$$\mathcal{P}_l : \mathcal{W}+ \rightarrow \mathcal{W}+ \tag{1}$$

$$\delta w = \mathcal{P}_l(w) \tag{2}$$

$$w^{'} = w + \delta w \tag{3}$$

The latent mapper trained individually for a specific semantic edit task, such as "Long Hair", "Cat face" or "Goat Beard" and image pair. The latent mapper is a shallow, densely connected neural network and is relatively quick to train.

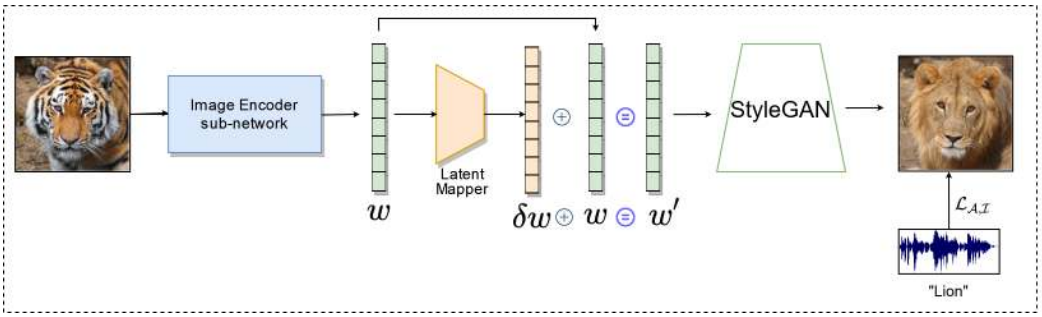


Figure 3: Architecture of Audio-StyleCLIP

The augmented $w^{'}$ vector is passed through an image decoder network to obtain the output stylized image. We employ StyleGAN2 [14] as the image decoder in our work. The stylized image is used to compute the loss function as described in Section 6.2.

The proposed system architecture is visualised in Figure 3.

An important detail to note here is that the loss ($\mathcal{L}_{(\mathcal{A},\mathcal{I})}$) depends on the audio input of the style description. In order to compute this loss, we make use of a recent work, namely AudioCLIP [15], which is an extension of the CLIP architecture for audio. As also discussed in Section 2, the Audio head of the AudioCLIP network is trained on environmental sounds dataset we do not expect it to produce reliable embedding for the speech input, and thereby using it may not provide a didactic estimate of loss. Hence we explore an alternate technique, described in further detail in Section 5.1, to AudioCLIP that is conducive to semantic speech processing. We study the characteristics and performance of both techniques.

5.1 Audio encoder sub-network architecture

We attempt to develop an audio encoder sub-network similar in function to that of the AudioCLIP that transforms an input audio signal into a latent embedding w_A in the CLIP latent space $\mathcal{W}+$, illustrated in Figure 4. We adopt an extended neural network that specializes in semantic speech processing called SpeechBERT [16]. We use the latent embedding produced by the SpeechBERT network and use a simple dense neural network to transform the embedding to map the semantic speech onto the CLIP latent space; we refer to this transformation network as the Audio Style Transform network. Figure 5 illustrates the architecture of the Audio Encoder sub-network.

6 Network Training

6.1 Datasets

We will use the Flickr-Faces-HQ (FFHQ) dataset [17]. The FFHQ dataset has high-resolution images (i.e., 1024 x 1024) crawled from Flickr, as the name suggests. It consists of a wide range of face images that include a variety of races, genders, ages, backgrounds, etc. It is also diverse regarding the accessories observed on faces like sunglasses, spectacles, scarfs, etc. The images were cropped, and unnecessary parts were pruned to get clear face images. The images from this dataset are used as target images for our Audio-StyleCLIP.

We also need styling input in the form of speech prompts, and as we do not have a readily available dataset of speech prompts, we attempt to generate an equivalent speech dataset using a Text-to-Speech conversion FastSpeech network.

We source the text prompts necessary for the text-to-speech conversion from the StyleCLIP dataset, which is, in turn, is adopted from the CLIP paper. We also use a similar prompt engineering trick to analyze zero-shot CLIP performance to augment the text prompts with equivalent semantic statements. This is accomplished by using a combination of prompts generated from a list of 80 templates and 1000 classes [9].

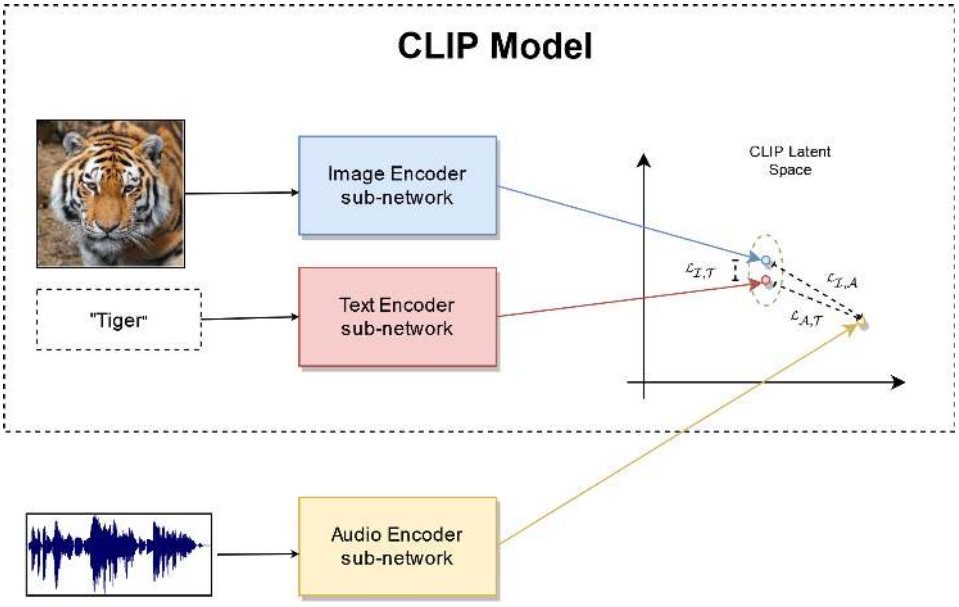


Figure 4: Latent Space mapping of Audio encoder sub-network

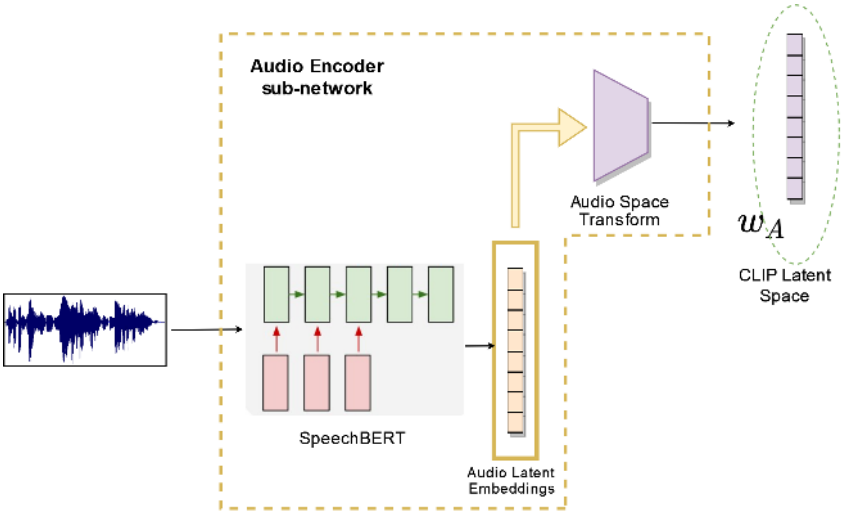


Figure 5: Architecture of Audio Encoder sub-network

6.2 Losses

The losses that we use in this work are identity loss $\mathcal{L}_{\mathcal{ID}}$, embedding loss $\mathcal{L}_{(\mathcal{A},\mathcal{I})}$ and $\mathcal{L}_2$ loss. The identity loss is used to ensure that the target image and the transformed image are close in terms of contrast, hue, other facial features, etc. It is as:

$$\mathcal{L}_{\mathcal{ID}} = 1 - \langle \mathcal{R}(\mathcal{G}(w^{'})), \mathcal{R}(\mathcal{G}(w)) \rangle \tag{4}$$

The embedding loss is the cosine distance between the embeddings of speech and image in the latent space which ensures closeness between transformed image and text command. It is as:

$$\mathcal{L}_{(\mathcal{A},\mathcal{I})} = \mathcal{D}_{(\mathcal{A},\mathcal{I})}(w, s) \tag{5}$$

Finally the L_2 loss is the L_2 norm between w and w_t. This too keeps in check the similarity between input and output images. It is:

$$\mathcal{L}_2 = ||w - w^{'}||_2 \tag{6}$$

Finally, the total loss will be:

$$\mathcal{L}_{total} = \lambda_{\mathcal{ID}}\mathcal{L}_{\mathcal{ID}} + \lambda_{(\mathcal{A},\mathcal{I})}\mathcal{L}_{(\mathcal{A},\mathcal{I})} + \lambda_2\mathcal{L}_2 \tag{7}$$

where,
$w^{'}$: Target image mapping in latent
w : Transformed image mapping in latent space
s : Speech prompt embedding in latent space
$\mathcal{G}$: pretrained StyleGAN generator
$\mathcal{R}$: pretrained ArcFace [18] [19] model for face recognition and the operator < . , . > is a dot product which gives a similarity between its arguments $\lambda_{\mathcal{ID}}$, $\lambda_{(\mathcal{A},\mathcal{I})}$, λ_2 : coefficients for loss importance

We expect that the

6.3 Training Routine

We follow a three-step training routine. First, we generate the synthetic dataset for the image-speech tuple in the first step, as described in Algorithm 1. It is followed by Audio encoder sub-network training using the synthesized dataset from the first algorithm to train the network (see Algorithm 2). Finally, Audio-SytleCLIP is trained to obtain a stylized image from a speech prompt for a given tuple, detailed in Algorithm 3.

Algorithm 1 Text-Speech synthesis

Require: Given the data $\{I_i, t_i\}$, where I_i and t_i are image and text captions repsectively.
Ensure: Generate $s_i = FastSpeech.forward(t_i)$, using a pretrained $FastSpeech$ network.

Algorithm 2 Speech-head training

Require: Using the synthesized data$\{I_i, s_i\}$, untrained network $AudioCLIP$ with parameters θ and loss $\mathcal{L}_{\mathcal{A},\mathcal{I}}$;
Ensure: $\min_\theta \mathcal{L}_{\mathcal{A},\mathcal{I}}(I_i, s_i; AudioCLIP)$

7 Novelty, Advantages and Limitations

While we are focusing on facial images, this can be extended to any image set, which can enable a lot of different style transfers tasks by giving speech commands. Some advantages are:

- More convenience than text input.

Algorithm 3 Audio-StyleCLIP training

Require: Pretrained network $AudioCLIP$, untrained network $LatentMapper$ with parameters ϕ, $GAN_Decoder$ and dataset set $\{I_i, s_i\}$, where I_i and s_i are images and speech prompts.

Ensure:

for every epoch **do**

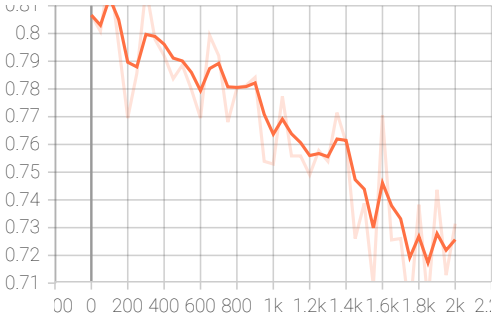
with no_grad:

$$w_{s,i} = AudioCLIP.speechEmbedding(s_i)$$
$$w_{I,i} = AudioCLIP.imageEmbedding(I_i)$$

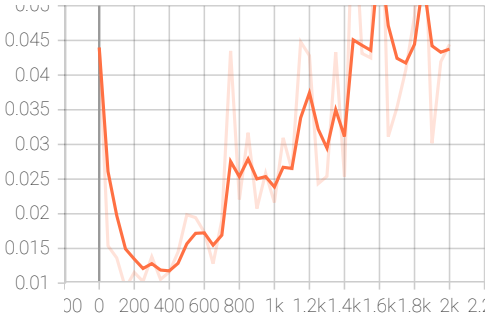
$w' = LatentMapper(w_{I,i}) + w_{I,i}$

$I_{out} = GAN_Decoder(w_{out})$

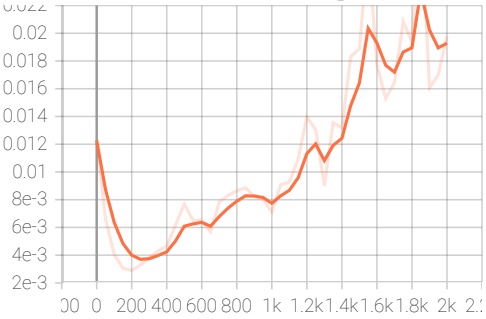
$\min_\phi\{\mathcal{L}_{total}(I_{out}, I_i, w_{s,i})\}$



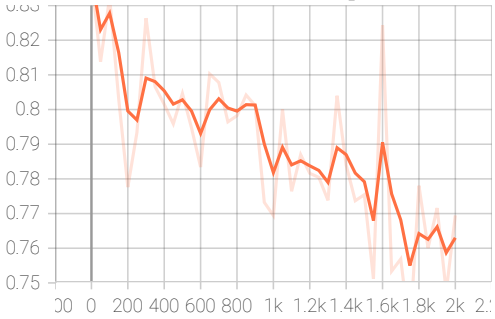
(a) CLIP loss v/s number of epochs



(b) ID loss v/s number of epochs



(c) L2 loss v/s number of epochs



(d) Combined training loss v/s number of epochs

Figure 6: Losses

- Enhanced abilities for artists, architects, designers, filmmakers (when extended to videos), and the like.
- This functionality can be incorporated in digital assistants like Siri, Alexa, etc., going ahead.

And some limitations are:

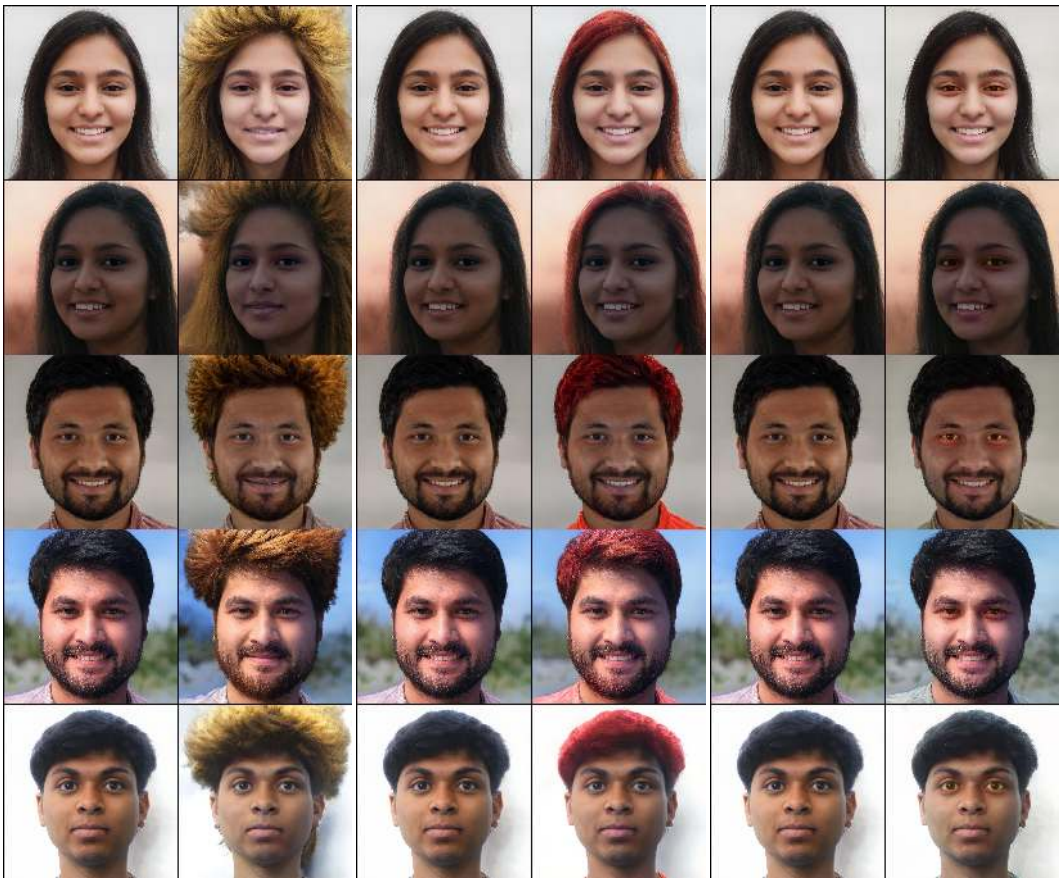
- The method we use AudioCLIP might not perform as well as we expect it to as AudioCLIP is trained on environmental sounds like vehicle sounds, natural sounds, etc. We believe that it might not interpolate well in our application, as we will have human speech data.
- The speech commands given for testing might be different from those that we give as input for the training of our Audio-Style CLIP in terms of pitch, frequency, noise components, etc. That might also deteriorate the performance.
- The latent mapper has to be trained on each (audio, image) pair. This makes the process computationally heavy.

8 Results and Discussions

The learning curves are as expected, $\mathcal{L}_{\mathcal{ID}}$ and $\mathcal{L}_2$ are increasing as the face starts transforming to something different from original face. The CLIP loss reduces as the image starts transforming to something close to the text prompt and so does the overall training loss.

Besides our results, we also highlight specific challenges we faced and would also report on our work in-order to resolve them. First of all, there was no available network for contrastive audio-image pairs(a one that would resemble OpenAI's CLIP for zero-shot image-text pair identification). For this reason, we initially decided to create our contrastive pretraining network that would mimic CLIP. However, the training problem faced high computational cost and short time-constraint in the project. Therefore, we abandoned the idea and used AudioCLIP, a contrastive audio-image-text pair pretraining network that works only for environmental sounds. Moreover, we thoroughly tested the styled image performance based on audio prompts, as most often, the audio prompts did not belong to human speech, and only a tiny fraction of training images were human portraits.

Some of our results on our own faces are as in Figure 7a, Figure 7b and Figure 7c. In the first case, the model tried to add a golden mane to the images and tried to make the faces look more serious. In the second one, it tried to change the hair colour to orange and also tried to change the colour of clothes to orange as it might be trying to consider clothes as the body of the tiger. In the last one, if seen closely, it changed the colour of the cornea in the eye from white to yellow and added red bags around eyes making it look more like a reptile's.



(a) Prompt 'Lion' (b) Prompt 'Look like a tiger' (c) Prompt 'Reptile eyes'

Figure 7: Styled Images

9 Conclusion

In summary, we were able to implement an audio prompt-based stylizing network. The network provides an alternative to text-based prompts, increasing the overall functionality of the StyleGAN networks. It also enables research in the StyleGAN community, providing insights on how speech-based prompts would modulate the style of a portrait.

Future work for the research in this direction include training contrastive audio-image pairs for human speech prompts and portraits. Stylized videos instead of images is another possibility.

References

[1] M.-y. Qin, Y.-c. Fan, B.-l. Liu, and X. Ma, "A survey of style transfer based on generative adversarial network," in *AOPC 2021: Novel Technologies and Instruments for Astronomical Multi-Band Observations*, vol. 12069. SPIE, 2021, pp. 314–320.

[2] C. Zhao, "A survey on image style transfer approaches using deep learning," in *Journal of Physics: Conference Series*, vol. 1453, no. 1. IOP Publishing, 2020, p. 012129.

[3] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *German conference on pattern recognition*. Springer, 2016, pp. 26–36.

[4] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, vol. 28, pp. 262–270, 2015.

[5] A. Jabbar, X. Li, and B. Omar, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Computing Surveys (CSUR)*.

[6] T. Karras, S. Laine, and T. Aila. (2018) A style-based generator architecture for generative adversarial networks.

[7] C. Troy. Neural style transfer with latte art, part 1 Layer depth. [Online]. Available: https://towardsdatascience.com/drawing-architecure-building-deep-convolutional-gans-in-pytorch-5ed60348d43c

[8] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 2085–2094.

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[10] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending CLIP to image, text and audio," *CoRR*, vol. abs/2106.13043, 2021. [Online]. Available: https://arxiv.org/abs/2106.13043

[11] ——, "Esresne(x)t-fbsp: Learning robust time-frequency transformation of audio," *CoRR*, vol. abs/2104.11587, 2021. [Online]. Available: https://arxiv.org/abs/2104.11587

[12] [Online]. Available: https://coqui.ai/

[13] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "Fastspeech: Fast, robust and controllable text to speech," *CoRR*, vol. abs/1905.09263, 2019. [Online]. Available: http://arxiv.org/abs/1905.09263

[14] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "Stylegan2 distillation for feed-forward image manipulation," in *European Conference on Computer Vision*. Springer, 2020, pp. 170–186.

[15] A. Guzhov, F. Raue, J. Hees, and A. Dengel. (2021) Audioclip: Extending clip to image, text and audio.

[16] Y.-S. Chuang, C.-L. Liu, H.-Y. Lee, and L.-s. Lee, "Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering," *arXiv preprint arXiv:1910.11559*, 2019.

[17] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[18] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[19] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for stylegan image manipulation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.