Spring 5-20-2020

Video Synthesis from the StyleGAN Latent Space

Lei Zhang

Video Synthesis from the StyleGAN Latent Space

A Project Presented to

The Faculty of Department of

Computer Science San José State

University

Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the

Degree of Science

By

Lei Zhang

April, 2020

The Designated Project Committee Approves

the Master's Project Titled

Video Synthesis from the StyleGAN Latent Space

By

Lei Zhang

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

April 2020

Dr. Christopher Pollett	Department of Computer Science
Dr. Leonard Wesley	Department of Computer Science
Dr. Philip Heller	Department of Computer Science

Acknowledgement

I would like to express my sincere gratitude and appreciation to Dr. Chris Pollett for his guidance and support throughout the entire project. It was my honor to have him as my advisor.

I would like also to take this opportunity to thank the committee members, Dr. Philip Heller and Dr. Leonard Wesley, for their suggestions and time.

Finally, I am grateful to my friends and family for always being a strong support for me throughout my career.

ABSTRACT

Generative models have shown impressive results in generating synthetic images. However, video synthesis is still difficult to achieve, even for these generative models. The best videos that generative models can currently create are a few seconds long, distorted, and low resolution. For this project, I propose and implement a model to synthesize videos at 1024x1024x32 resolution that include human facial expressions by using static images generated from a Generative Adversarial Network trained on the human facial images. To the best of my knowledge, this is the first work that generates realistic videos that are larger than 256x256 resolution from single starting images. This model improves the video synthesis in both quantitative and qualitative ways compared to two state-of-the-art models: TGAN and MocoGAN. In a quantitative comparison, this project reaches a best Average Content Distance (ACD) score of 0.167, as compared to 0.305 and 0.201 of TGAN and MocoGAN, respectively.

Keywords - Generative Adversarial Network (GAN), video generation, StyleGAN, 3D convolutions

TABLE OF CONTENTS

1 INTRODUCTION

Realistic video synthesis helps to reduce the cost and time required to produce videos and moreover eases transferring facial expressions and body actions to a different person. The development of Generative Adversarial Networks (GANs) [4] has enabled video synthesis through two competitive neural networks, where the first learns how to generate fake data while the other learns how to identify fake data (see Section 2 for more detail on GANs). This project attempts to synthesize videos that human cannot easily distinguish between those which are fake and real.

Since their invention by Ian Goodfellow in 2014 [4], GANs became highly successful in image synthesis, video generation, object detection, etc. and have been used as a machine learning model to synthesize videos [2][5][6][7]. However, GANs cannot generate a video clip that has notable differences from the training dataset, and there is currently no research towards generating a video without mimicking the same actions from GAN training videos. In this project, I propose a model to generate videos of human emotions using randomly generated human faces.

Two state-of-the-art models of video synthesis in paper [5] and [7] have attempted using a separate temporal layer to improve the performance of video generation with GANs, which improved both the quality and efficiency of video synthesis. The limitations are: 1) it is difficult to generate high-resolution videos and 2) more time is required to generate videos.

Generative Adversarial Networks establish the latent space after training, which is a representation of compressed data. This project proposed a model generates high-resolution videos of human facial expressions directly from a pre-trained StyleGAN [22] latent space which contains a compressed representation of human facial images. Unlike traditional methods of using GANs to generate videos, this project uses them to generate images and then finds potential frames in the image GANs' latent space. By utilizing the development of StyleGAN, this project

generates high-resolution, arbitrarily long, and realistic videos of human facial expressions.

Many video generation researchers seek to find a whole model to directly generate videos similar to the generative ability of image GANs [5][6][7]. However, these attempts usually cannot directly use pre-trained image GANs' latent space. In addition, the models that function well in image generation cannot directly be used for video generation due to the required extra temporal layer to learn in both discriminators and generators. This project separates image generation from video generation.

The organization of this report is as follows: The background chapter provides information on image GANs, video GANs, embedding images into the StyleGAN latent space, face emotion prediction, etc. The implementation chapter explains all the stages of generating a video from a pre-trained StyleGAN latent space, and the experiments chapter describes the dataset to be used and the results. Finally, I discuss the findings and conclusions in the conclusions chapter.

2 BACKGROUND

Generative Adversarial Networks consist of two neural networks, the generator and the discriminator, which compete with each other. The generator takes noise vectors as the input and outputs fake data similar to the training dataset. The discriminator takes two inputs of the generator output and a training dataset to decide whether these data are real or fake. The generator strives to "fool" the discriminator so that it cannot distinguish between generator-produced fake data and real training data, while the discriminator aims to detect fake data as well as possible. During this process, the generator can produce fake data as close as to the training dataset as possible.

In the following sections, I introduce the convolutional neural networks (CNNs), development of Image GANs and video GANs. CNNs are common layers in both GANs and Very Deep Convolutional Networks (VGGs) [25]. Moreover, embedding images into the StyleGAN latent space is discussed in this section since they represent the key mechanisms for video synthesis in this project. Sequence prediction was used to predict emotions from YouTube movie trailers. Furthermore, I discuss VGG16 which is used as an image features extractor.

2.1 Convolutional Neural Networks (CNNs)

A Convolutional Neural Network (CNN) is a neural network with one or more convolutional layers. Regular CNNs can handle spatial dimensions, which has been proved suitable for recognizing images and extracting their features. They are called 2D CNNs since they cannot present more than 2 dimensions. Images include width and height, which is a typical 2-dimensional (2D) space, while videos include 3-dimensional (3D) space with an extra time dimension. Therefore, regular CNNs include limitations for understanding videos which include temporal information to learn.

Ji et al. [29] are the first to propose 3D CNNs that extends a convolutional layer from 2D to 3D which can learn temporal features. Instead of using a square convolutional filter, 3D CNNs

use a cube which can extract features for the third dimension (times). The emergence of 3D CNNs has resulted in many attempts to use GANs with 3D CNNs to generate synthetic videos [6], and 3D CNNs can be applied in both the discriminator and the generator or separately.

2.2 Image GANs

The pre-trained StyleGAN latent space is used in this project, and therefore it is important to understand how StyleGAN was developed in order to understand the latent space. The Progressive growing GAN concept is adopted by StyleGAN to generate high-resolution images and is introduced as well.

2.2.1 Progressive growing GAN

Progressive growing GANs (ProGANs) [9] achieve some of the best results in generating high-resolution images (e.g., 1024x1024). Before this paper's proposal of ProGANs, there were no papers able to use GANs to generate high-quality images.

The authors suggest firstly training the GAN on lower resolution images before gradually increasing the resolution of generated images by adding new layers, because the training of low-resolution images helps in training higher-resolution images. Figure 1 shows the ProGAN gradually increasing the image resolution in the training to address the challenge of GANs' generation of high-quality images.

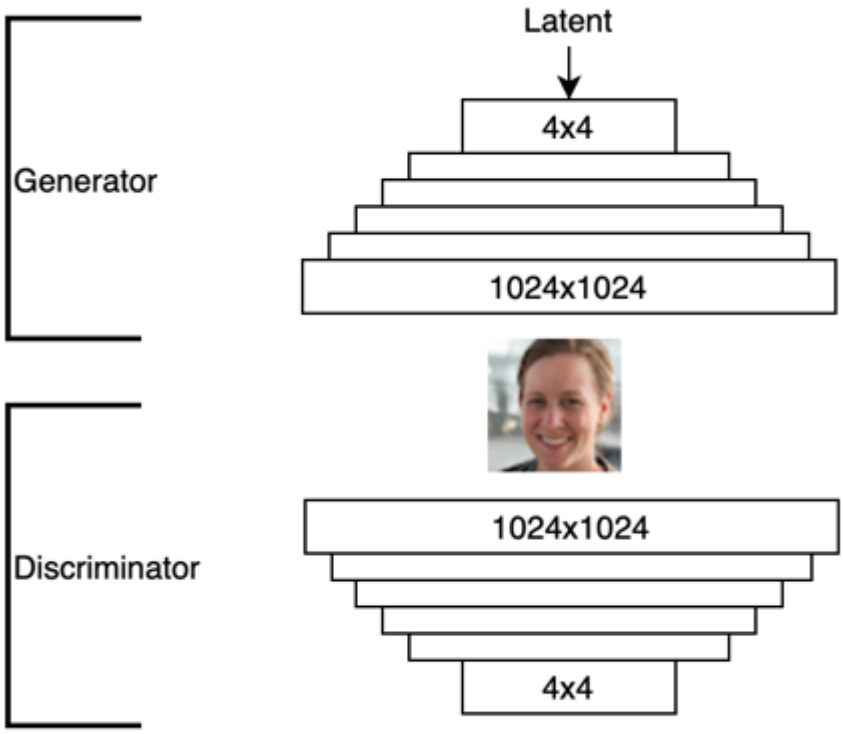

Figure 1: ProGANs model. In order to generate a high-resolution image, both the generator and the discriminator begin from a low-resolution image first, and then the model gradually increase the resolution of generated images.

2.2.2 StyleGAN

StyleGAN [22] uses the progressive growing concept as a baseline and improves it to achieve better style mixing. Style mixing in StyleGAN means it is able to learn attributes such as freckles, hair style and face shape, and therefore enable transfer them to other images in the StyleGAN latent space. StyleGAN introduced a new dataset of human faces called Flickr-Faces-HQ (FFHQ). The current StyleGAN commonly generates images with blobs that look like water splotches, as shown below in Figure 2:

Figure 2: StyleGAN generated an image with a water droplet-like artifact in the top-right corner.

A common GAN uses random noise vectors as its input layer Z. StyleGAN omits the traditional input layer Z and creates a mapping network to generate an intermediate latent space W. The authors introduce a function called AdaIN (Adaptive Instance Normalization) [22] which transfers the input vector W into generated images. Figure 3 shows the quality of random images generated by StyleGAN.

Figure 3: StyleGAN random generated images.

2.2.3 StyleGAN2

Karras et al. [20] further improved the image synthesis quality of StyleGAN. This paper represents the new state-of-the-art model of image GANs and fixed the water-splotches issue in StyleGAN. Furthermore, through the redesigned generator and loss function that measures deviates from training data, StyleGAN2 provides favorable results for video interpolation. Figure 4 shows the quality of images generated by StyleGAN2.

Figure 4: StyleGAN2 random generated images. The images are sharper and with less distortion compared to Figure 3.

2.2.4 Face Pose Synthesis

Face Pose Synthesis is similar to this project of generating videos from image GANs' latent space. For face pose synthesis, Souza et al. [21] propose a conditional learning with GAN which labels the faces with different pose positions from -75° to 75°, and can recover face pose using these pose labels. However, this method requires re-training the latent space and is limited to pose generation.

2.3 Video GANs

3D CNNs were widely used in GAN to generate videos. Vondrick et al. [6] propose an idea to separate the static background and dynamic foreground by designing two streams of generators, where the foreground generator creates dynamic moving while the background generator produces static scenes. Their model is called Video GAN (VGAN) and assumed that a video always has a static background and thus cannot generate videos with dynamic backgrounds. In this paper, both discriminator and generator use 3D CNNs.

Although using 3D CNN GANs seems relatively intuitive as an approach to generate videos, the quality of videos produced by this approach tends to be low. 3D CNNs have caused overfitting and inefficient training problems [10]. Many similar ideas have combined a temporal layer with 2D CNNs to replace 3D CNNs. Pascanu et al. [10] propose a method of replacing 3D CNNs with 2D CNNs which has improved both the performance and accuracy of video classification. Saito et al. [2] explored a new network that uses Temporal Generative Adversarial Nets (TGAN) to acquire time features, which combined a 1D and a 2D generator to learn both spatial and temporal features that are similar to a 3D CNN GAN. However, TGAN still uses 3D convolutional layers in its discriminator. TGAN claimed a better result compared to the VGAN model which uses 3D CNNs int both its discriminator and generator.

Clark et al. [7] introduce a model that uses two discriminators to learn spatial and temporal features respectively. This model combines gated recurrent unit (GRU) and ResNet for temporal layer learning, and although it still used 3D CNNs in its discriminatory, it also introduced a separate discriminator solely for learning spatial features of images. This paper significantly improved the quality of generated videos with up to 256 X 256 resolution and up to 48 frames, and the goal was to learn spatial and temporal separately in order to improve the performance of video generation.

Tulyakov et al. [5] propose a model called Motion and Content Decomposed GAN (MoCoGAN), which is similar compared to the study in [6] that separates the motion and static scene. This model preferred a recurrent neural network (RNN) to learn motion features rather than a 3D discriminator and moreover used a 2D GAN to generate a sequence of frames instead of a 3D GAN generator used in [6]. This reduced the complexity of training 3D CNNs and delivered better results compared to both [6] and [2]. Finally, the 2D GAN model prevents introducing more variations with the third dimension in a 3D GAN and thus represents one of the best approaches that can generate plausible video clips.

2.4 Sequence Prediction

I used the sequence prediction techniques in this section to generate possible emotion sequence. A long short-term memory (LSTM) model was used in my project, however, I introduce recurrent neural networks (RNNs) first to better understand how LSTMs function.

2.4.1 Recurrent Neural Networks (RNNs)

Traditional neural networks cannot learn a sequence of data, meaning they cannot remember the previous data. RNNs attempt to address this issue using loops and make decisions using not only current data but also previous data. Figure 5 shows the input x_t as a sequence of data and the output as h_t.

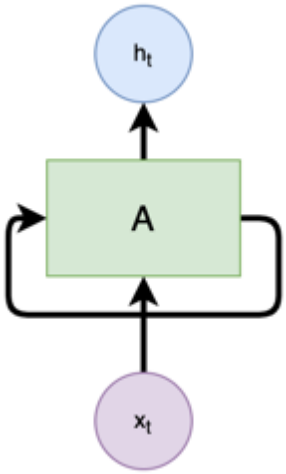

Figure 5: Recurrent Neural Network.

2.4.2 Long Short-Term Memory (LSTM)

Recurrent Neural Networks have been successfully used in many areas such as image captioning and language modeling, however they cannot learn long-term dependencies due to vanishing gradient problems [1]. Hochreiter et al. [3] propose an LSTM mechanism which regards a special kind of RNNs to address this problem.

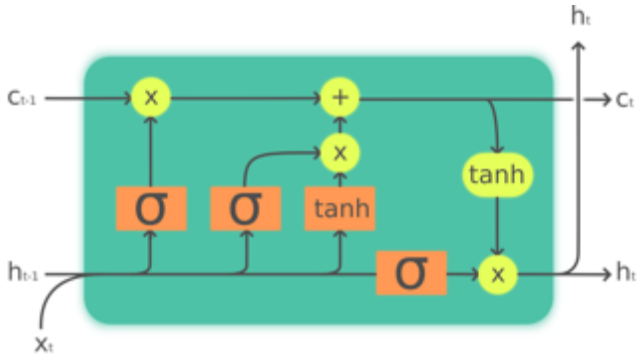

Figure 6: LSTM Block.

LSTM is designed to learn temporal features in a sequential dataset. In video classification, it is commonly used to learn the relationships among a sequence of frames. In Figure 6, C, X and h represent the memory, input, and output respectively, where the h_{t-1} denotes a previous hidden state (output) while C_{t-1} similarly denotes previous memory. LSTM uses cell states that can selectively remove or add information, and the output depends on three inputs: current input, previous output and previous hidden state. Therefore, it can predict a result based on remembering a sequence of training data.

The sigmoid function is usually used as an activation function in gates. Gates are LSTM internal functions that only allow optionally data to come in. Through these mechanisms, LSTM acts like a human in order to selectively remember and forget things for a long sequence. There are three gates in LSTM: forget gate, input gate and output gate, which represent the primary mechanisms of regulating the flow of information.

- **Forget Gate**

It makes a decision regarding which information should be kept or dropped while taking both the current input and the previous hidden state as its input.

- **Input Gate**

It decides which state to transfer to the cell state.

- **Output Gate**

It generates the next hidden state, which also forms the input of the next LSTM block.

2.5 Very Deep Convolutional Networks (VGG)

I use Keras VGG16 function in this project to extract image features, which were used to compare the difference between two images when embedding an image to the latent space. Simonyan et al. [25] propose a very deep convolutional networks (VGG) for image recognition and suggests using small (3x3) convolution filters instead of larger ones, which results in improved image classification. The proposed VGG network has 16 to 19 weight layers.

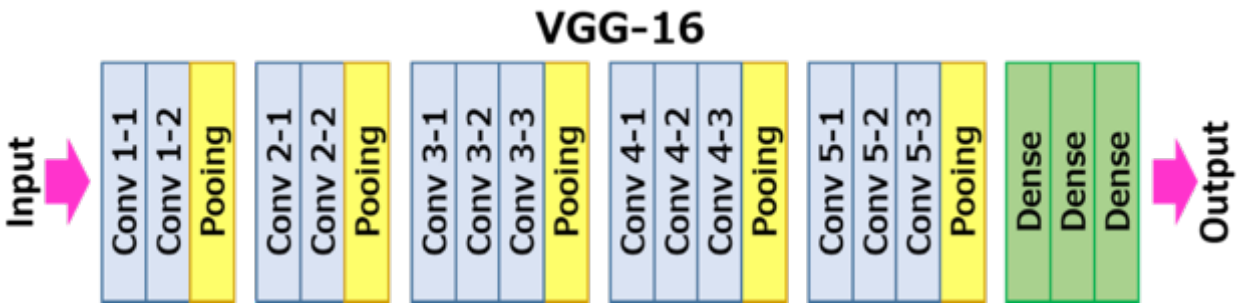

Figure 7: VGG-16 layer definition.

VGG16 has 16 weight layers which includes 13 convolutional and 3 dense layers. In Figure 7, Conv 1 and Conv 2 have 64 and 128 filters respectively, Conv 3 has 256 filters, and Conv 4 and Conv 5 both have 512 filters. In this project, in order to get a feature map but not classes of images

in this project, I do not use the final output layer but instead cut off the training and use the output from Conv 5.

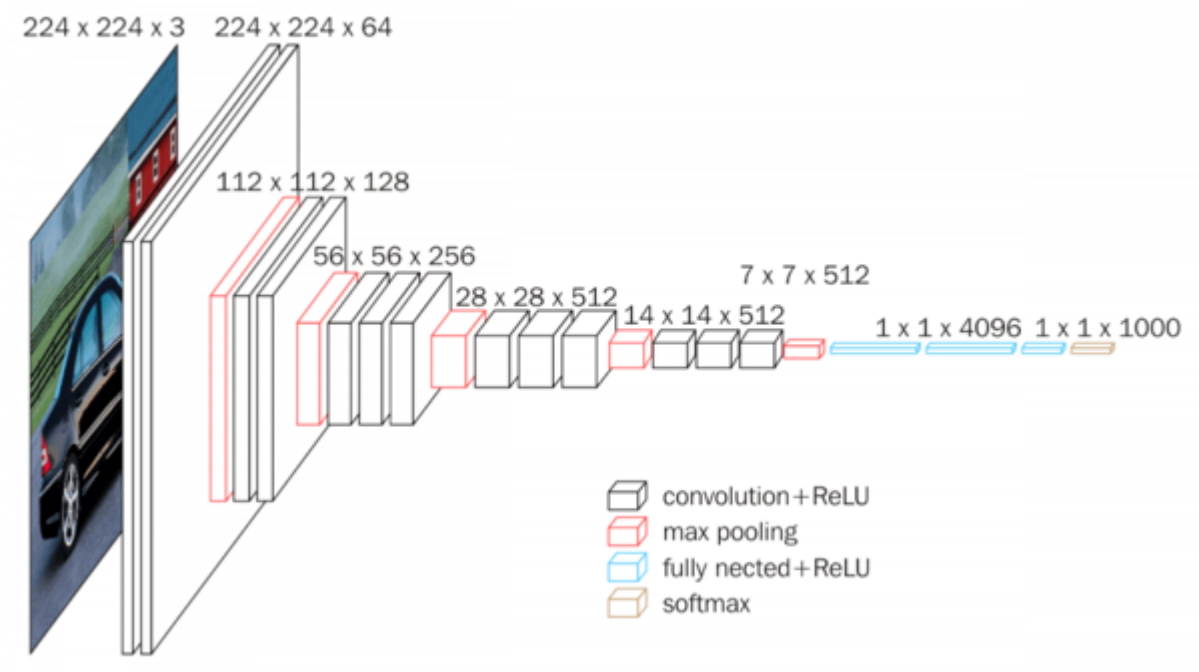

Figure 8: VGG-16 architecture.

Figure 8 shows the architecture of VGG16, which takes RGB images with fixed 224x224 pixels as input. In the paper, the authors achieved 92.7% top-5 accuracy with VGG16 that trained on the ImageNet dataset. The output of VGG16 has 1000 classes.

2.6 Deep Residual Learning (ResNet)

In this project, I use ResNet-50 to predict mappings from an image to a noise vector. VGGs represents an effort to make deep neural networks deeper. However, He et al. [28] mentioned that deep neural networks face the degradation problem that additional layers cause lower training accuracy which is not caused by overfitting.

He et al. [28] addressed the degradation problem using a framework shown below in Figure 9. As a result, ResNet achieves 3.57% error training on the ImageNet dataset which is better than VGG nets. ResNet-50 is one of the deep residual networks which has 50 layers.

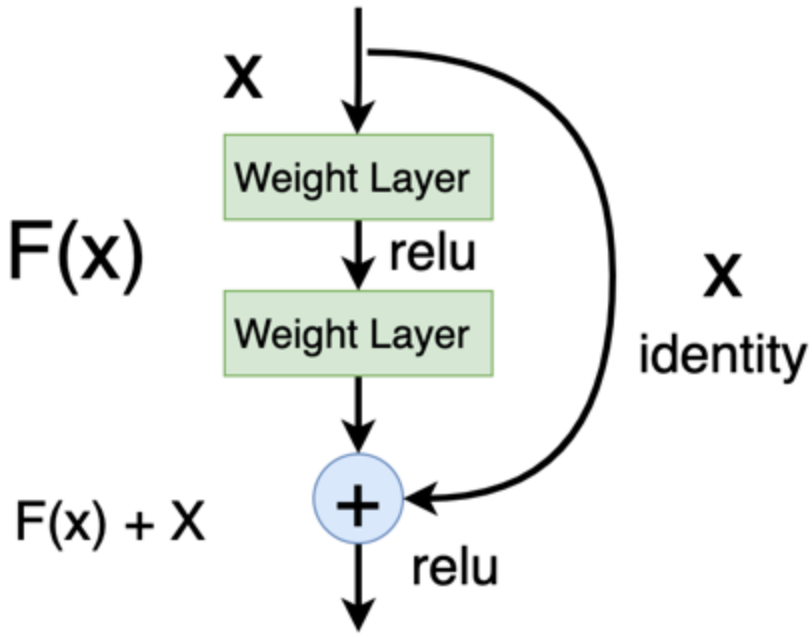

Figure 9: Residual learning.

2.7 Embed Images into Latent Space

2.7.1 Precise Recovery of Latent Vectors from GANs

Latent vectors (codes) are latent variables which map the data into GANs latent space. Lipton et al. [24] propose a gradient-based method to recover images from a latent space that transfers the invert problem to a directly gradient optimization problem, as shown in Figure 10. Suppose z is a noise vector which can produce image ϕ, and we want to find z′ which generates an image as close as ϕ. Although this method functions well for finding an image that is generated from the given latent space, it fails when choosing a random image.

$$z' \leftarrow \text{clip}(z' - \alpha \nabla_{z'} ||\phi(z) - \phi(z')||_2^2).$$

Figure 10: Recovering a latent vector [24].

2.7.2 Image2StyleGAN

Abdal et al. propose Image2StyleGAN [17] that is able to recover a random image from a StyleGAN latent space. In Figure 11, the authors propose a loss function to compare an original image and a generated image, allowing an optimized latent code to be found to best represent the original image. This technology enables transferring a video clip into a pre-trained StyleGAN latent space.

$$w^* = \min_w L_{percept}(G(w), I) + \frac{\lambda_{mse}}{N}\|G(w) - I\|_2^2 \quad (1)$$

$$L_{percept}(I_1, I_2) = \sum_{j=1}^{4} \frac{\lambda_j}{N_j}\|F_j(I_1) - F_j(I_2)\|_2^2 \quad (2)$$

Figure 11: Image2StyleGAN loss functions [17].

2.8 Noise Vector Arithmetic and Video Interpolation

A GANs' latent space contains lots of useful information. We can use a random noise vector to map an image in the latent space, however, it is difficult to search the latent space and acquire desired images. Noise vector arithmetic can help to understand how the latent space is constructed. Furthermore, that helps to add certain features on top of an image.

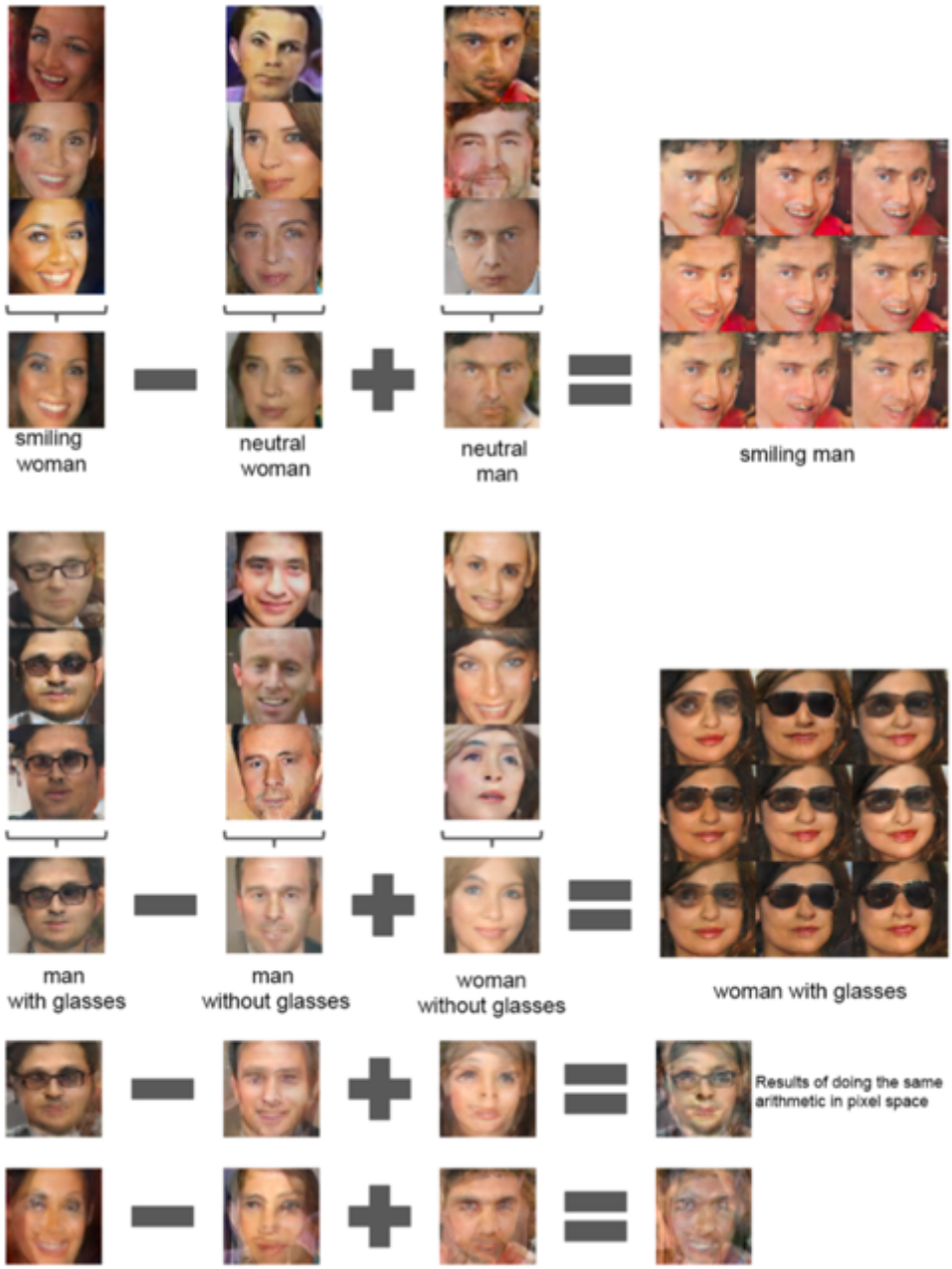

Figure 12: Vector arithmetic [19].

Radford et al. [19] showed that it is possible to generate meaningful images using only vector arithmetic in a pre-trained latent space. Figure 12 indicates that a smiling woman subtracts a neural woman and adds a neural man to generate a smiling man. In addition, the interpolation between latent vectors generates plausible in-between frames, as shown below in Figure 13. Video interpolation means generate a sequence of transition images between two giving frames.

Figure 13: Interpolation between two images [19].

3 IMPLEMENTATION

The main goal of my video generation technique is to find the frames of a potential video inside a pre-trained image GANs' [20][22] latent space. This project more specifically intends to create a video of a person displaying different emotions in a latent space trained with human faces.

In the StyleGAN latent space, a noise vector has 18x512 dimensions, which is an enormous space to search. There are two challenges to resolve in this project: 1) Search the latent space to find the same person with meaningful emotions and 2) generate intermediate frames to connect all emotions.

There are two ways to approach the first challenge: coarse mapping and fine mapping. Coarse mapping uses the style transfer directly from manipulate the latent code from another image. As proposed in paper [19], we can add two noise vectors to generate another meaningful image, however this method fails to generate meaningful emotions with the same person since either the face or emotion do not align with the expected result. Fine mapping creates directions for a few emotions and adds them to the target person by adjusting a coefficient number. This method works well, which allows generating different emotions on the same person with clear images. More details are discussed in the section below. The fine mapping requires training datasets labeled with face emotions, and for this purpose I use both the MUG facial expression database and IMPA-FACE3D database.

I employed two methods to address the second challenge: a linear search and binary search. Given two vectors with one as the start point and the other as the end point: 1) A linear search evenly fills in the desired number of vectors in between and 2) a binary search converges faster to close to the end point. I found no significant difference the two methods, possibly because the start and end images are too similar to be notice differences.

3.1 Datasets

3.1.1 FFHQ

StyleGAN Flickr-Faces-HQ (FFHQ) is a human faces dataset which consists of 70,000 high-quality PNG images at 1024×1024 resolution. These aligned images were downloaded from Flickr and were used to train StyleGAN.

3.1.2 IMPA-FACE3D

This project uses the database IMPA-FACE3D to train emotion directions. The dataset collects 534 static images from 30 people with 6 samples of human facial expressions, 5 samples of mouth and eyes open and/or closed, and 2 samples of lateral profiles.

3.1.3 MUG Facial Expression Database

The MUG facial expression database [23] regards another dataset of facial expressions and consists of 86 subjects and 6 basic expressions: anger, disgust, fear, happiness, sadness and surprise. Each video has a rate of 19 frames, and each image has 896x896 pixels. I cropped and scaled the image to 1024x1024 pixels.

3.1.4 CelebA Dataset

The CelebFaces Attributes Dataset (CelebA) [27] contains more than 200K celebrity face images and is used by this project to test the image embedding performance.

3.1.5 YouTube Movie Trailers

In order to predict emotions, I use another dataset which includes random picked movie trailers were downloaded from YouTube.

3.2 Model Overview

There are three stages to training the model: Stage 1 creates directions of key frames which represent different emotions; Stage 2 predicts the emotion sequence in movie trailers; and Stage 3 replays the emotion sequence to another human face. After these stages, this model can generate a high-resolution video clip.

The first stage has three steps:

1. Embed all the IMPA-FACE3D images to the StyleGAN latent space and output 534 latent codes mapping to 534 images. A pre-trained VGG network is used to extract image features.
2. Generate a training dataset with latent codes and facial expressions labels which are the output of Step 1.
3. Train a logistic regression model to predict directions of human facial expressions. Logistic regression is a classification machine learning model to predict binary results, which is an extension of linear regression.

In Stage 2, I use a pre-trained classification model to extract all of the faces within the current video. A LSTM model is used to predict a random length of sequence for emotions. These predicted emotion sequences form the input of a dataset for the next stage of generating a video.

Finally, Stage 3 transfers the predicted emotions to a random human face in order to compose a video. A random human face was generated in the StyleGAN latent space. I use the directions in Stage 1 to generate all the emotions as the keyframes, where the larger the coefficient number the larger the human emotion. Finally, I create a linear function to fill in transition frames between each two adjacent emotions before finally creating a video with the same person displaying different emotions.

3.3 Face Alignment

Face alignment determines the location of the human faces and crops faces to the center of canvas in an image. Without face alignment, the model cannot recover faces with accurate details and therefore cannot learn a proper direction of emotions. I use the StyleGAN original face alignment code to perform face alignment on all the training datasets, except I changed the fill function from “reflect” to “edge” when using NumPy to pad the image. After the alignment, the output is 1024x1024 images with human faces in the center.

3.4 YouTube Video Preprocessing

I followed the following steps for processing the video:

1. Extract all videos to frames
2. Use CNN-based face detector to detect faces in frames
3. Only keep the frames include human faces
4. Align and crops frames to 1024 x 1024 resolution

3.5 Emotions Prediction

EmoPy is a python tool which predicts emotions by providing images of people’s faces. It uses CNNs to classify and detect at most seven emotions using four convolutional layers and two pooling layers. The final full connected layers generate a classification output.

3.6 LSTM Emotion Sequence Prediction Model

I use LSTM to predict a sequence of emotions using pre-processed YouTube video frames. There are four LSTM layers in the model, and I added dropout on each layer, as shown in Figure 14. The output dense layer has seven classes which represent seven emotions, while the input data regarded a sequence of integers which represent emotions that require preprocessing to keep only emotion changes. The purpose of this model is to predict a possible emotion change, and the output is a list of emotions. This step enables generating an arbitrary length of video.

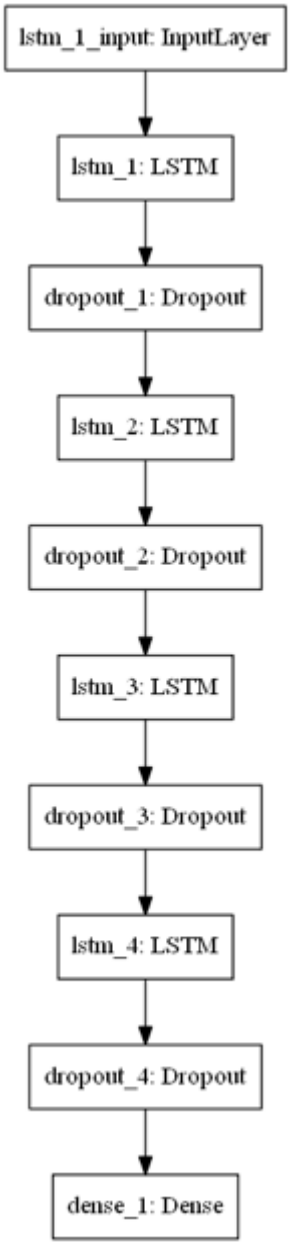

Figure 14: LSTM model for emotion sequence prediction.

3.7 VGG16

As shown in Figure 15, I use the Keras VGG16 model to extract the image features where the model has been pre-trained with the ImageNet dataset. Keras is a Python deep learning library.

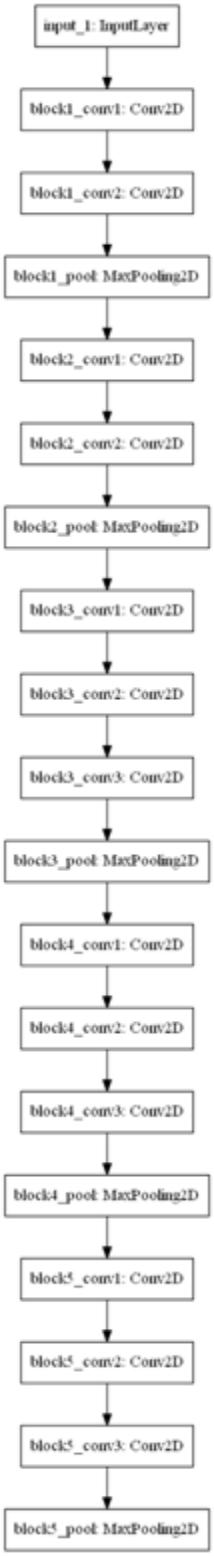

Figure 15: VGG16 model for image features extraction.

3.8 Image Reconstruction from the Latent Space

A further step involves mapping the MUG facial expression database and IMPA-FACE3D database to the StyleGAN latent space. Figure 16 shows the way of backpropagating gradients through the generator model. Instead of updating each layer's weight, this reconstruction process only updates the latent code while the weights of the neural networks receive no changes.

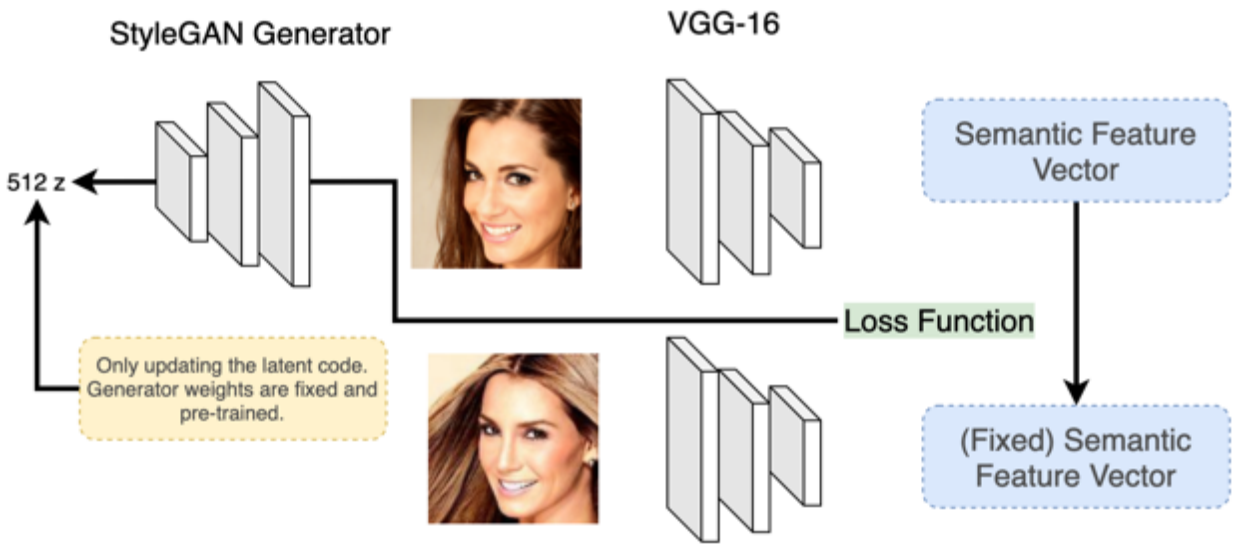

Figure 16: Embed images into StyleGAN latent space.

3.9 Generate Keyframes with Latent Space Manipulation

Once the directions have been trained, the model learns to apply the direction of emotions to any person that is generated in the latent space. I use the following steps to generate keyframes:

- Pick a face in the random generated samples with StyleGAN2 latent space.
- Add the face's latent code with emotion directions with a coefficient number, which is fixed based on the experience but can be improved by optimization from learnings.
- Apply masks to the generated keyframes. A latent code has 18 vectors, and the mask layer overwrites the last 13 vectors to keep the face without significant changes.
- Save the latent codes on all directions for the next step.

3.10 Interpolation in the Latent Space

This step attempts to generate intermediate frames between any two keyframes, which is also called inbetweening and provides a smooth transition from one image to another. There are 32 intermediate frames generated between two keyframes in this project.

4 EXPERIMENTS AND RESULTS

In this project, I use the StyleGAN latent space, which is pre-trained with Flickr-Faces-HQ dataset that generates images at 1024 x 1024 resolution. The StyleGAN latent space was trained using only static images which lack any sequence of video frames.

4.1 Coarse Image Recovery from Latent Space

This method uses ResNet50 to directly predict noise code by providing images. The training dataset can be easily generated from the StyleGAN latent space and includes pairs of latent codes and generated images. As Figure 17 indicates, this method can predict images in the latent space on a real time basis but cannot generate the exact same images as the input.

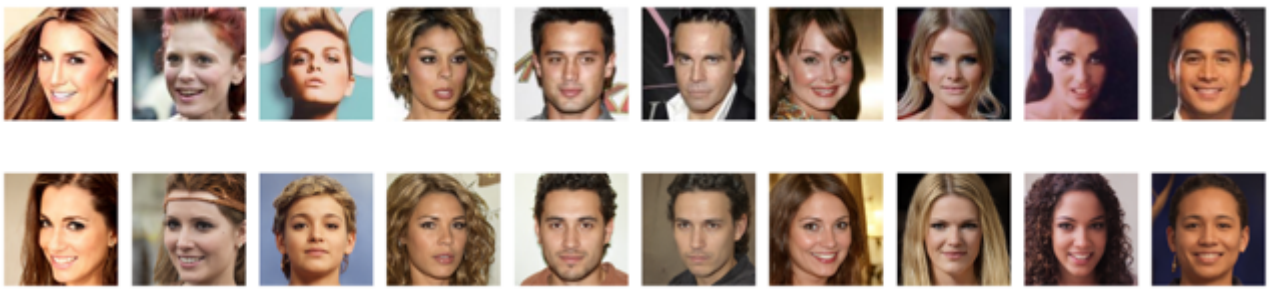

Figure 17: ResNet50 generated image pairs. Line 1 includes the input images and Line 2 has the predicted images.

4.2 Fine Images Recovery from the pre-trained Latent Space

This method requires training a neural network to learn a latent vector mapping of an image. Figure 22 shows the concept functions well, where line 1 contains random images from the CelebA dataset while line 2 shows recovered images from the StyleGAN latent space. All the images were precisely recovered from the pretrained latent space, although they did not train the dataset using any of these images. The limitation of this method is that it requires more time to train. In Figure 18, I trained the model for 1,000 epochs for each image.

Figure 18: Embed CelebA images in the StyleGAN latent space.

4.3 Mimic Face Pose without Training Directions

This method transfers face poses to a random generated face in the StyleGAN latent space. Only a single video is needed to perform the transition, and it also does not require training a model to learn the direction of face poses, which saves significant computation resources. As shown in Figure 19, I use a mask for transfer learning only the first five vectors in the latent space and only the first 256 dimensions in the first five vectors. However, it is difficult to transfer learn the emotions using the same method since the emotions are not linear distributed across the 18 latent vectors in StyleGAN. Therefore, I must train a model to learn the directions of each emotion.

Figure 19: Transfer face pose without learning directions. Line 1 and 3 are the original frame, while line 2 and 4 are the transferring frames.

4.4 Transfer Facial Attributes to Another Person

Another method is to use labeled data to train model to acquire the directions of each emotion, which allows directly add them to the target latent vectors. This method functions well with a coefficient of 8 in the experiments.

The StyleGAN latent code includes 18 of 512-dimension vectors. I use only the first 8 while the last 10 vectors are unchanged, which ensures that the face does not change too much. Figure 20 below shows the difference between adding a mask and no mask to generate emotions.

Figure 20: Generated images with mask and without mask. Line 1 and 3 are with a mask, while line 2 and 4 are without a mask.

In Figure 24, the person shows more unexpected changes when without a mask. The purpose of the mask is to retain the original face when transferring emotions. I found the first 8 vectors control the change of emotions, which means we can leave the other 10 vectors unchanged.

I use a linear interpolation method to generate all the frames in the latent space among all keyframes. This allows creating all transfer frames in between to give the movie a smooth

appearance. The limitation is that there are some artificial variations in generated human facial expressions.

Another attempt involved recovering a full video in the pre-trained StyleGAN latent space, and then I generated a random latent code as the start frame to predict a video. Once acquiring the start latent code, I use the same trend from the latent codes recovered from the training video to predict a new video clip. This method does not appear stable, so I abandoned this direction.

4.5 Predict Emotions from YouTube Video Clips

The model predicts seven emotions: calm, anger, happiness, surprise, disgust, fear and sadness. I build a model with four LSTM layers to predict the emotions in YouTube videos, and using LSTM to predict emotion sequence result in 85% accuracy. In Figure 21, I use 0 to 6 to represent the 7 emotions in the experiments.

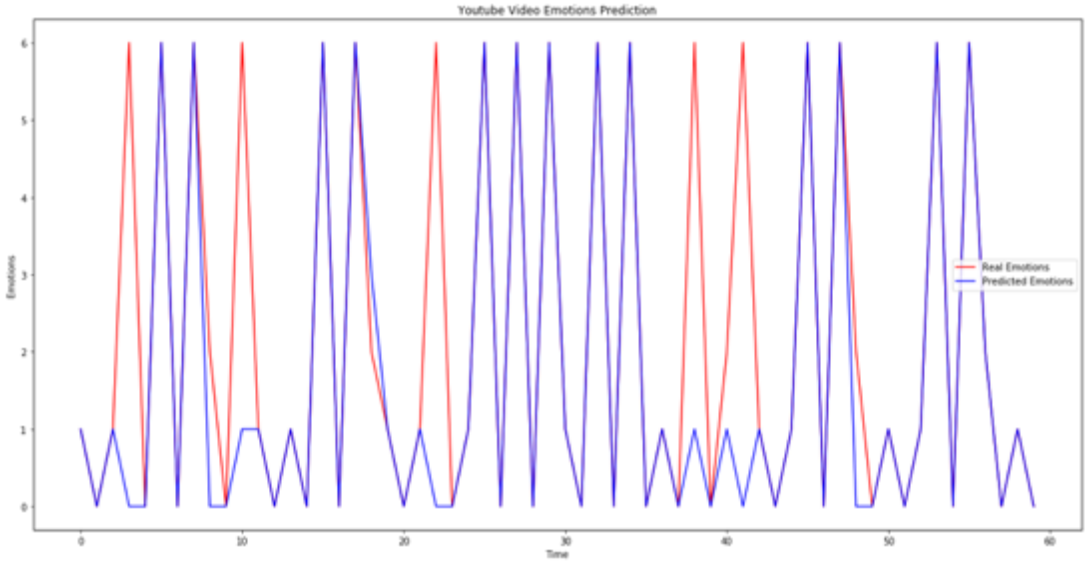

Figure 21: Emotion sequence prediction after training.

After training, this model predicts a sequence of emotions, all of which can be mapped to the StyleGAN latent space to generate video clips with a synthesis face.

4.6 Video Synthesis

I choose a random generated human faces in the experiments to demonstrate that the video interpolation results.

The video at 1024x1024x32 resolution is shown in Figure 22 below:

Figure 22: Generated video frames. Full video available here.

4.7 Compare result with TGAN and MocoGAN

4.7.1 TGAN

Figure 23 shows the result from TGAN with the UCF101 dataset. The generated videos are shown at 64x64 resolution.

Figure 23: TGAN-generated video with UCF101 dataset [2].

4.7.2 MocoGAN

The generated emotions from MocoGAN are shown in Figure 24 below at 96x96 resolution.

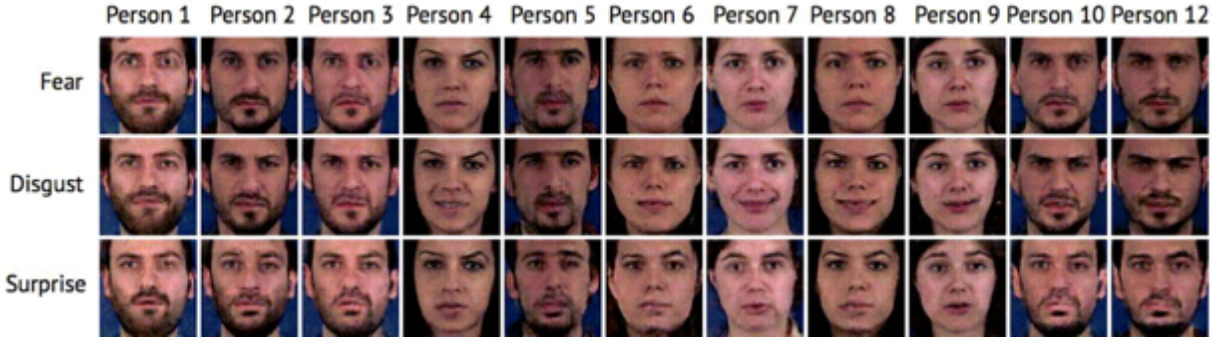

Figure 24: MocoGAN-generated video with MUG Facial Expression Database [5].

4.7.3 My Model

This project generates six face expressions at 1024x1024 resolution with three random generated faces and I additionally generated three frames of the six expressions. In Figure 25, 26 and 27, I picked three frames for each emotion which represents facial emotion transition.

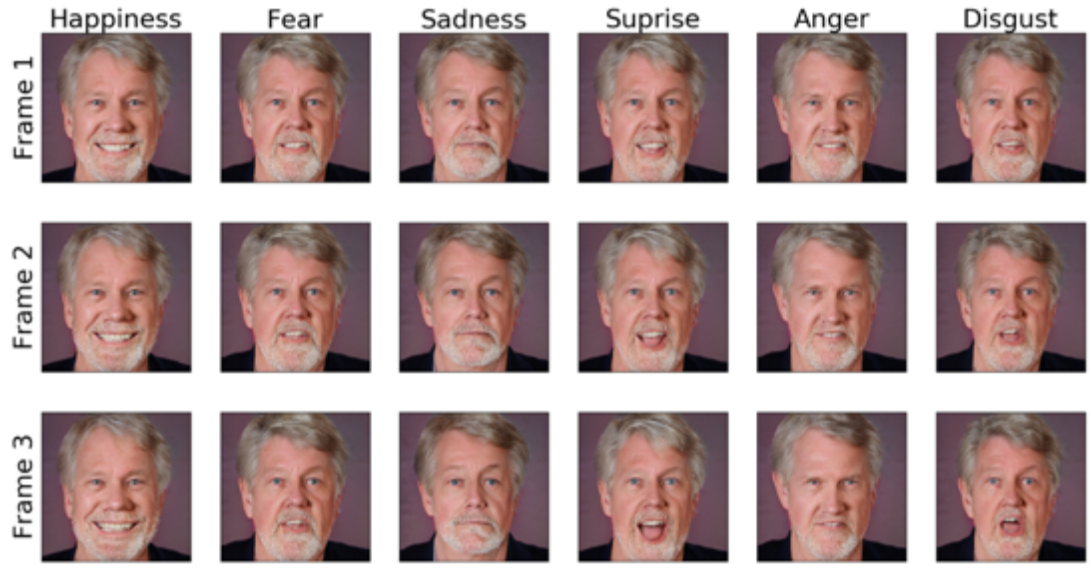

Figure 25: Facial Expressions with IMPA-FACE3D Database.

Figure 26: Facial Expressions with IMPA-FACE3D Database.

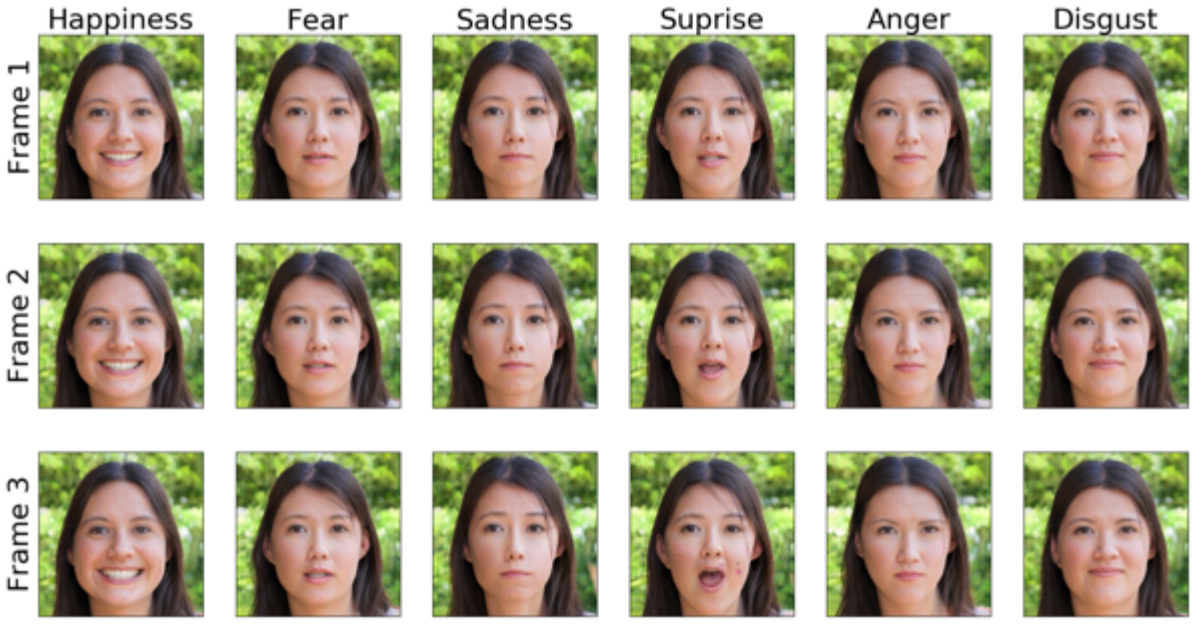

Figure 27: Facial Expressions with IMPA-FACE3D Database.

4.8 Quantitative Comparison

I used the Average Content Distance (ACD) [5] metric to measure content consistency of a generated video. The ACD is calculated with average L2 distance among all consecutive frames in a video. A smaller ACD score is better which means a generated video is more likely to be the same person. I generated 210 videos using 35 randomly generated faces with each face having the same 6 different facial expressions. As shown in Table 1, my model shows the best result in generating consistent facial expressions compared to TGAN and MoCoGAN.

ACD	Facial Expressions
TGAN [5]	0.305
MoCoGAN [5]	0.201
My Model	0.167

Table 1: Video generation content consistency comparison.

I asked 100 workers from Amazon Mechanical Turk (AMT) which video looked more realistic. As shown in Table 2, most of the participants answered that the video generated by this project looked more realistic compared to the other two models. All of the videos used in the comparison were generated by training with the MUG facial expression database. The videos in my model have a 1024x1024 resolution, while TGAN and MoCoGAN have a resolution of only 128x128 because their models do not generate high-resolution videos.

User preference, %	Facial Expressions
My Model / TGAN	90 / 2
My Model / MoCoGAN	77 / 3

Table 2: Video generation preference.

5 CONCLUSIONS

Due to the limitations in using the GANs method to generate videos, I propose a new method to directly synthesize videos from a pre-trained image GANs latent space. I chose the StyleGAN latent space due to its success in upscaling the image resolution, however, this method can be used in any image GANs latent space. The results show that this method not only improves the speed of video generation with transfer learning from image GANs, but also that it is suitable for generating high-resolution video clips. To the best of my knowledge, there is currently no GANs able to generate video clips with 1024x1024 resolution. Furthermore, using a pre-trained 2D image latent space with well-selected video frames, we may predict better videos in the future. There are two potential directions for continuing this project: 1) predict the directions in the latent space with self-labeled data; 2) random video synthesis with proper loss functions.

The limitation of this project is that it relies on aligned images and I found it difficult to properly recover the frames of a video without image alignment. The reconstructed video from the aligned frames are not as smooth as the original due to frame shakes following the alignment of each image.

Acknowledgments

Figure 6, Figure 7 and Figure 8 are licensed under a free license and were downloaded from Wikipedia. All the other figures, images and tables in this report were created by me except the ones have references.

REFERENCES

[1] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks* 5.2, 1994.

[2] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping, " In *ICCV*, 2017.

[3] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural computation 9.8*, 1997.

[4] Goodfellow, Ian, et al., "Generative adversarial nets," Advances in neural information processing systems, 2014.

[5] S. Tulyakov, et al., "Mocogan: Decomposing motion and content for video generation," In *CVPR*, 2018.

[6] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[7] A. Clark, J. Donahue, and K. Simonyan, "Efficient video generation on complex datasets," *arXiv preprint arXiv*:*1907.06571*, 2019.

[8] P.Isola, et al., "Image-to-image translation with conditional adversarial networks," In *CVPR*, 2017.

[9] T. Karras, et al., "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[10] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," In *ICML*, pages 1310–1318, 2013.

[11] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions calsses from videos in the wild,"Technical Report CRCV-TR-12-01, *UCF Center for Research in Computer Vision*, 2012.

[12] S. Xie, et al., "Rethinking spatiotemporal feature learning for video understanding," *arXiv preprint arXiv:1712.04851*, 2017.

[13] M. Rohrbach, et al., "A Database for Fine Grained Activity Detection of Cooking Activities," In *CVPR*, 2012.

[14] L. Gorelick, et al., "Actions as space-time shapes," In *TPAMI*, 29(12):2247-2253, 2007.

[15] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[16] A. Clark, J. Donahue and K. Simonyan, "Adversarial video generation on complex datasets," *arXiv preprint arXiv*:1907.06571, 2019.

[17] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?," *Proceedings of the IEEE International Conference on Computer Vision*. 2019.

[18] P. Bojanowski, et al., "Optimizing the latent space of generative networks," *arXiv preprint arXiv*:1707.05776, 2017.

[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv*:1511.06434, 2015.

[20] T. Karras, et al., "Analyzing and improving the image quality of stylegan," *arXiv preprint arXiv*:1912.04958, 2019.

[21] D. M. Souza, and D. D. Ruiz, "GAN-Based Realistic Face Pose Synthesis with Continuous Latent Code," *The Thirty-First International Flairs Conference*, 2018.

[22] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[23] N. Aifanti, C. Papachristou, and A. Delopoulos, "The MUG facial expression database," *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10. IEEE*, 2010.

[24] Z. C. Lipton, and S. Tripathi, "Precise recovery of latent vectors from generative adversarial networks," *arXiv preprint arXiv*:1702.04782, 2017.

[25] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv*:1409.1556, 2014.

[26] Y. LeCun, et al., "Object recognition with gradient-based learning," *Shape, contour and grouping in computer vision*. Springer, Berlin, Heidelberg, 1999.

[27] Z. Liu, et al., "Deep learning face attributes in the wild," *Proceedings of the IEEE international conference on computer vision*, 2015.

[28] K. He, et al., "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[29] S. Ji, et al., “3D convolutional neural networks for human action recognition,” *TPAMI*, *35(1):221– 231*, 2013.