Facial Expression Video Synthesis from the StyleGAN Latent Space

Lei Zhang
San Jose State University
San Jose CA, USA
lei.zhang01@sjsu.edu

Chris Pollett
San Jose State University
San Jose CA, USA
chris.pollett@sjsu.edu

May 14, 2021

Abstract

Generative neural network models have been used to create impressive synthetic images. However, artificial video synthesis is still hard, even for these models. The best videos that generative models can currently create are a few seconds long, distorted, and low resolution. We propose and implement a model to synthesize videos at $1024 \times 1024 \times 32$ resolution that include human facial expressions by using static images generated from a Generative Adversarial Network trained on human facial images. To the best of our knowledge, this is the first work that generates realistic videos that are larger than 256×256 resolution from single starting images. Our model improves video synthesis in both quantitative and qualitative ways as compared to two state-of-the-art models: TGAN and MocoGAN. In a quantitative comparison, we achieve a best Average Content Distance (ACD) score of 0.167, as compared to 0.305 and 0.201 for TGAN and MocoGAN, respectively.

Keywords: GAN, StyleGAN, video synthesis, facial expression, latent space

1 Introduction

Realistic video synthesis helps to reduce the cost and time required to produce videos by easing the process of combining videos and permitting one to do operations like transferring facial expressions and body actions to and from one person to a different person. The development of Generative Adversarial Networks (GANs) [4] is one technique commonly used for video synthesis. It performs video synthesis through the use of two competitive neural networks, where the first learns how to generate fake data while the second learns how to identify fake data (see Section 2 for more detail on GANs). In this paper, we present our GAN-based model to synthesize videos that are hard for humans to detect are fake.

Since their invention by Ian Goodfellow in 2014 [4], GANs have become a very popular tool for image synthesis, video generation, object detection, etc. They have also been used as a machine learning model to synthesize videos [10][11][12][3]. However, GANs cannot generate video clips that have notable differences from their training dataset, and there is currently little research towards generating videos that do more than just mimic the underlying actions from the training videos. Our proposed model tries to address this by using a pre-trained GAN to generate videos of humans carrying out various emotions. What is interesting is that we start from randomly generated human faces and then generate videos of emotion shifts and emotion combinations which are not in the underlying data set.

Two state-of-the-art models of video synthesis [11] and [3] have attempted to use a separate temporal layer to improve the performance of video generation with GANs. These models improved both the quality and efficiency of video synthesis. However, their techniques were limited in that they are difficult to use to generate high-resolution videos and they are more time consuming than previous state-of-the-art techniques for video generation.

The training process of a Generative Adversarial Networks establishes a latent space for the domain being trained. In our case, this space is a compressed representation of the underlying video or image. Our model generates high-resolution videos of human facial expressions directly from a pre-trained StyleGAN [6] latent space which contains a compressed representation of human facial images. Unlike traditional methods of using GANs to generate videos, we use them to generate images and then find potential frames in the image GANs' latent space. By utilizing StyleGAN in this way, we generate 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 [11][12][3]. However, these attempts usually cannot directly use a 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 layers that would be need for both the discriminators and the generators. Our model allows one to separate image generation from video generation and leverage the former in the training of the latter.

The organization of this paper is as follows: The first section provides background information about image GANs and video GANs. It describes techniques to embed images into the StyleGAN latent space and describes face emotion prediction techniques. The next section explains the stages of our model for generating a video from a pre-trained StyleGAN latent space. The experiments section describes the datasets pre-processing done, explains our experimental designs, and gives our results. Finally, the last section draws some conclusions concerning our model and our results.

2 Related Work

Video generation is not a simple task. Our work takes advantage of recent research in four areas: image prediction, video prediction, latent space image embedding, and scripted animation. In this section, we describe prior work in these areas.

Image prediction: Since we use a pre-trained StyleGAN latent space as a starting point for our model, it is important to understand how StyleGAN was developed in order to understand this latent space. StyleGAN makes use of the Progressive GAN technique to generate high-resolution images, so we need to introduce this technique as well. Progressive GANs (ProGANs) (Karras et al. [5]) achieve some of the best results in generating high-resolution images (e.g., 1024×1024). The Karras et al. paper was also the first to describe using GANs to generate high-quality images. To achieve their results, the authors suggest first training GANs on lower resolution images and then gradually increasing the resolution of generated images by adding new layers. The idea is that the training on lower resolution images will help in the training of the higher-resolution images. Figure 1 shows how this process works. StyleGAN [6] uses the progressive increase in image size idea as a starting point and improves on it to allow one to mix in different "styles". Here styles are attributes such as freckles, hair cut, and face shape. StyleGAN enables the transfer of these styles to other images in the StyleGAN latent space. The StyleGAN paper also introduced a new dataset of human faces called Flickr-Faces-HQ (FFHQ).

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) [6] which transfers the input vector W into generated images. Karras et al. [7] further improves upon the image synthesis quality of StyleGAN and represents the current new state-of-the-art model for image GANs. The original StyleGAN suffered from the frequent appearance of water-splotch-like artifacts which is fixed in this revised model. Furthermore, it uses a redesigned generator and loss function that better measures deviation from training data. This latter makes StyleGAN more suitable for video interpolation.

Video prediction: 3D CNNs are a widely used building block for GANs used to generate videos. Vondrick et al. [12] propose a video GAN model that separates the static background and dynamic foreground into two streams handled by different generators. A foreground generator is used to create dynamic movement while a background generator produces static portions of scenes. Their model is called Video GAN (VGAN). It assumes that a video always has a static background and so it cannot be used to generate videos with dynamic backgrounds. For each of their GANs, both the 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 are prone to overfitting and inefficient training problems [9].

Another approach to video generation is to combine a temporal layer with 2D CNNs instead of using 3D CNNs. Pascanu et al. [9] propose a method of replacing 3D CNNs with 2D CNNs that improved both the performance and accuracy of video classification. Saito et al. [10] explored a network that uses Temporal Generative Adversarial Nets (TGAN) to acquire time features and then combines 1D and a 2D generator to learn both spatial and temporal features. However, TGAN still uses a 3D convolutional layers in its discriminator. Saito et al. claim better results compared to the VGAN model which uses 3D CNNs in both its discriminator and generator. Clark et al. [3] introduce a model that uses two discriminators to learn spatial and temporal features respectively. This model combines agated recurrent units (GRUs) and ResNet for temporal layer learning. They still use 3D CNNs for their discriminator, however, they also use a separate discriminator solely for learning spatial features of images. This paper significantly improves the quality of generated videos with up to 256×256 resolution and up to 48 frames.

Tulyakov et al. [11] propose a model called Motion and Content Decomposed GAN (MoCoGAN). Like VGAN [12] it separates the motion and static parts of a scene, however, it uses a recurrent neural network (RNN) to learn motion features rather than a 3D discriminator. It also uses a 2D GAN to generate a sequences of frames rather than the 3D GAN generator used in VGAN. This reduces the complexity of training and delivers better results as compared to both [10] and [12].

Latent Space Image Embedding: Given a starting image and a trained image GAN network, one useful to perform operation is to find a latent vector input to the generator of the GAN that produces an image close to the starting image. Lipton et al. [8] propose a gradient-based method to recover images from a latent space that frames this inverse problem as a gradient optimization problem. Let ϕ be our target image, and suppose z is a noise vector that when sent through the generator produce image ϕ'. We can use a metric $m(\phi, \phi')$ that measures the distance between these images as a loss function for our optimization. Although this method works well for finding an image that is generated from the given latent space, it tends to fail for most choices of m when applied to a random image. Abdal, et al. propose a variation, however, on this idea, called Image2StyleGAN [1], that is able to recover a random images from the StyleGAN latent space. The authors propose a new loss function, a combination of perceptual loss and the pixel-wise MSE loss to compare an original image and a generated image that allowing an optimized latent vector to be found to best represent the original image. This technology enables one to easily transfer frames from a video clip into a pre-trained StyleGAN latent space.

Scripted Animation: One application for our video generation model is in the building of a naturalistic text to animation system. Translating text into animation is valuable for screenwriters and companies to prototype ideas and save labor. The latest text-to-animation systems assist users to visualize the overall look of a final animation quickly, but are not generally intended as a final polished animation. Zhang el al. [13] is a representative method to generate animations from screenwriting scripts that is based on Natural Language Processing (NLP) and an animation generation engine (much like a video game engine) which represents a typical architecture of existing text-to-animation systems. The idea is from text using NLP techniques to handle complex sentences, subjects and objects that need to be animated are determined. Using the animation engine with a library of pre-built 3D object models and animation snippets, appropriate objects are selected, placed in a scene, and animated. This approach is a fair bit different than the approach we experimented with using our neural net. In our system, a user supplies a start still image of a face and then in text gives a sequence of commands indicating how they would like the face to change. From this a video sequence is constructed using our neural network technique. No human built/scan representation for the face/object in the scene is needed.

3 Model

Our models were developed in Python using Kera and scikit-learn. There are three stages to training our model: The first stage is used to train a submodel that can generate emotion direction vectors in the StyleGAN latent space. The second stage trains, using movie trailers, a submodel to predict plausible, keyframe, facial emotion sequences from a starting face. Finally, the third stage trains a submodel that can, using our first submodel, replay an emotion sequence beginning from a starting human face. Finally, we can use these three submodels to generate a video from a random starting face by using the second submodel to generate a plausible emotion sequence and then using the third submodel to transfer this emotion sequence to the random starting face and interpolating in the latent space between these keyframes.

The first stage of our training in turn consists of three steps:

1. A pre-trained VGG-16 network is used to extract image features from all 534 IMPA-FACE3D images. VGG-16 is a 16-layer image classification model. We use its 10th layer output for our feature latent space. We then measure the similarity between an embedded IMPA-FACE3D image and the latent vector that results by taking a random noise vector, applying StyleGAN to it to generate an image, and then applying the same VGG-16 operation to this. Using back-propagation, we can train for noise vectors that correspond to a given feature. After the training, we get a list of 534 latent vectors which mapping to the 534 IMPA-FACE3D images. More details are discussed in Subsection 3.2.

2. Generate a training dataset consisting of (latent vectors, facial expression label) pairs using the output of the previous step.

3. Use this dataset to train a model that gives us emotion directions in the latent space. We tried both logistic regression and an SVM approach as techniques to do this. We ended up choosing the logistic regression model for our experiments as it had a shorter training time as compared to our SVM model. We trained our logistic model, $p(\vec{x}) = \frac{1}{1+e^{-(\beta_0+\vec{\beta}\cdot\vec{x})}}$, on a dataset of (latent face codes, facial expression) pairs. The trained vector $\vec{\beta}$ from our model predicts from an input face vector $\vec{x}$ whether or not that face expresses or does not express a given emotion label. These trained regression coefficients $\vec{\beta}$ could then be used to represent an emotion direction, and could be applied

linearly to a face vector $\vec{w}$ in the latent space to make a new face vector representative of $\vec{w}$ but expressing the given emotion more. We use an L_2 penalty for our logistic regression model, and a linear kernel for our SVM model. We use the default values from scikit-learn for the hyperparameters in these two models.

After the first stage of training is complete, the second stage uses a pre-trained face detection model EmoPy [2] to extract all of the faces within training data video clips. An LSTM model is then used to predict the emotions displayed by these faces during the video. These predicted emotion sequences form the input of training dataset for the last stage of our model.

Finally, for the last stage of training our model, we want to train how to transfer an emotion sequence to a random human face in order to compose a video. To do this a random human face was generated in the StyleGAN latent space. We use the directions in Stage 1 to generate all the emotions as the keyframes, where the larger the coefficient number the more exaggerated the human emotion. Finally, we 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.

In the following subsections, we discuss the particular techniques and technologies used in our model above. Subsection 3.1 introduces the face alignment preprocessing step we needed to train emotion directions. Then the subsection 3.2 gives an architecture overview of image embedding. The subsection 3.3 discusses how we generate keyframes. Finally, the subsection 3.4 discusses how we generate inbetweening frames.

3.1 Face Alignment

We use the term face alignment to describe the process of locating a human face in an image and producing a cropped, centered image of just the face. Face alignment allows us to have consistent images when training emotion directions in the StyleGAN. We used the StyleGAN project's original face alignment code to perform face alignment on all of our training datasets. The only change was that we switched the fill function from "reflect" to "edge" when using NumPy to pad images. After the alignment, the output data consisted of 1024×1024 images with human faces in the center.

3.2 Image Reconstruction from the Latent Space

For the first stage of our model, we need to map the MUG facial expression database and IMPA-FACE3D database to the StyleGAN latent space. Figure 1 shows how gradients backpropagate through our generator model to achieve this. Instead of updating each layer's weight, this reconstruction process only updates the latent vector while the weights of the neural networks receive no changes.

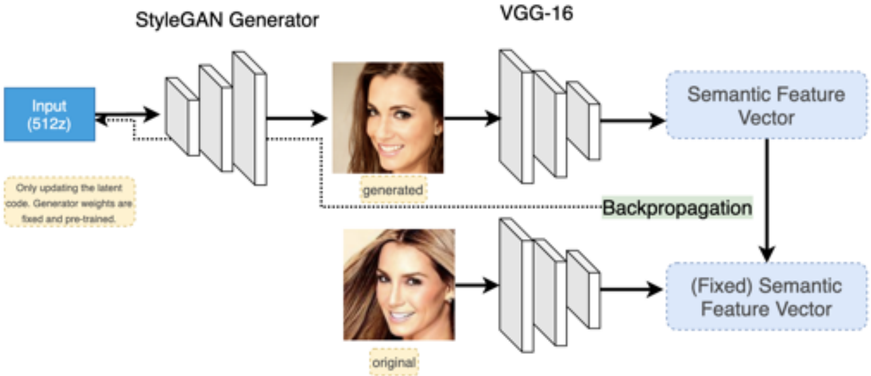

StyleGAN Generator
VGG-16
Input
(512z)
Only updating the latent code. Generator weights are fixed and pre-trained.
generated
Semantic Feature Vector
Backpropagation
original
(Fixed) Semantic Feature Vector

Figure 1: Embed images into StyleGAN latent space.

3.3 Generate Keyframes with Latent Space Manipulation

Once the directions have been trained, our model learns to apply emotion directions to any person that is represented in the latent space. This allows us to generate the keyframes of our video, and we use the interpolation to fill in the frames to represent the time lapse. We use the following steps to train for this:

- Pick a face in the random generated samples with StyleGAN latent space.
- Add to the face's latent vector a fixed scalar multiple of the emotion direction we are training for. This scalar hyperparameter can be tuned to adjust how strongly an emotion is expressed.
- Apply masks to the generated keyframes. A latent space vector is internally represented as a matrix of 18 row vectors each of 512 dimensions. The mask layer keeps the last 10 vectors unchanged from the original image vector and so only the first 8 are changed. StyleGAN takes 18 row vectors as the input of its synthesis network, please refer [6] for more details. We found this choice of vectors helps to preserve a person's identities. Roughly, the first 8 rows are responsible for emotions. We also found that using masks in the first 8 rows can further help to preserve a person's identities.
- Save the latent vectors on all directions for the next step.

3.4 Interpolation in the Latent Space

To animate between keyframes, we use interpolation in the latent space. An in-between latent vector $\vec{I}$ is computed as a vector sum $t\vec{K}_1 + (1-t)\vec{K}_2$ of two keyframe vectors. Here $t \in [0,1]$. The distance between keyframes affects how realistically the in-between latent vectors correspond to a face. Through trial and error, we settled on 32 intermediate frames between two keyframes for our experiments.

4 Experiments

We now describe the experiments we performed both to develop our video generator and to test its efficacy. Our first experiments were used to try to see how well we could transfer pose information onto a new face vector in the latent space. This was done using a transfer masking approach that we then modified in our final model after this original model failed when we tried to use it to transfer emotions in the StyleGAN's latent space. As part of learning what worked best, we experimented with both logistic regression and SVM models, so we describe our results for both. Given our ability to transfer pose and emotion information on novel faces, we then experimented with simple, scripted animations from a starting face. Finally, we conducted experiments using our whole proposed model to generate videos starting from single images. Training was done and experiments were conducted using a single desktop with a NVIDIA Titan RTX 24GB GPU.

4.1 Transferring Face Pose without Training Directions

As we mentioned earlier, a latent vector in StyleGAN is internally represented as a matrix of 18 row vectors each of 512 dimensions. One naive method for transferring some features from one face to another face is to fix most of these 18 vectors in the target face and copy the remaining vector from the face we want to transfer information from. We experimented with using this idea to transfer face poses to a random generated face in the StyleGAN latent space. One nice aspect of this approach is that it can be used with a single starting face and a video of poses of another face that we would like to transfer. It does not involves training of directions in the latent space, nor does it require much computation resources to carry out. Figure 2 shows the result of mask transfer learning where the first 256 dimensions of the first 5 vectors in the latent space are transferred from a video (lines in 1 and 3) to the latent starting vector of lines 2 and 4. For the case of changes of pose, this approach achieves semi-plausible results, however, when the same approach was tested for emotions, the results were not very satisfying. This is probably because emotions moves different parts of the face differently, which cannot be captured by such a simple linear transfer in the latent space.

4.2 Transferring Facial Attributes to Another Person

As we mentioned in the description of our model, the basic idea behind our approach is to train a classifier for a feature we would like to be able to transfer to a new face, then use the trained weights in this classifier as a direction vector $\vec{f}$ in the latent space. Given a new face latent vector $\vec{x}$ to apply the feature, we simply calculate $\vec{x} + t\vec{f}$ for some choice of real number t to get a latent vector that our generator will reconstruct to a face that has that feature more or less depending on sign and magnitude of t. For the StyleGAN latent space, and using logistic regression as our classification method, $t = 8$

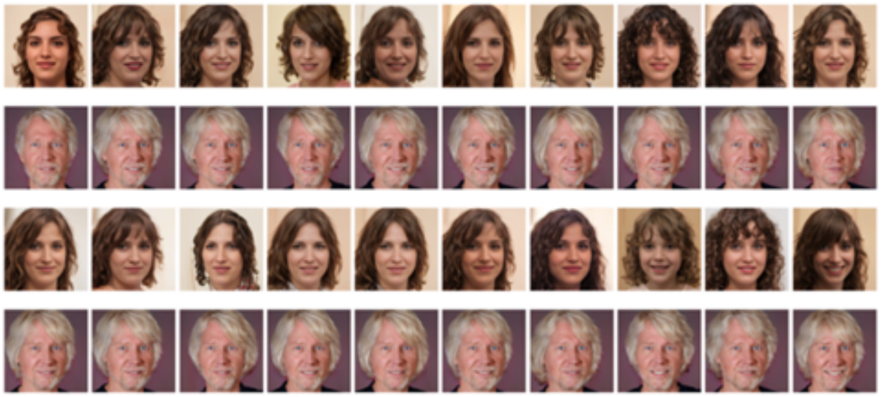


Figure 2: Transferring face pose without learning directions. Line 1 and 3 are the original frames, while line 2 and 4 are the generated frames. These frames consist of the first 5 vectors from the frames above it on lines 1 and 3 together with 13 vectors unchanged from the start frame of the given row.

seemed to work the best in our experiments. Figure 3 shows the effect of applying the “surprise” direction for different choices of t beginning from a starting face. I.e., $t = 0$.

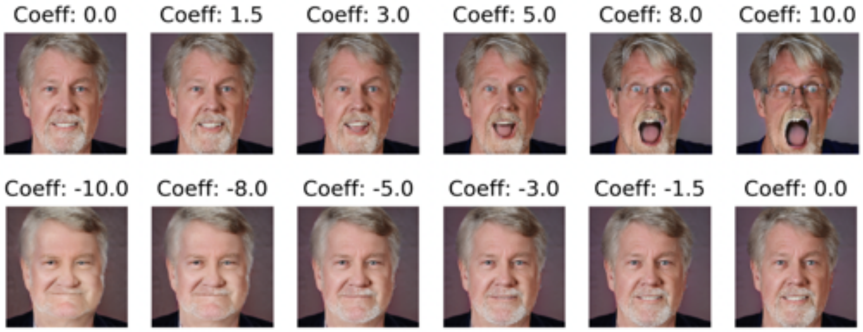


Figure 3: Linear Facial Expression Prediction - Surprise

The pure direction approach described above has a tendency to produce artifacts because too much of the original face changes as we add in more of a feature. To dampen the amount of change, we experimented with using masking as well as feature directions. Figure 4 shows the effect of using a mask that restricts the feature vector added to only the first 8 vector rows of the 18 vector rows of a StyleGAN vector. Lines 1 and 3 show with the mask, lines 2 and 4 without.

As can be seen in Figure 4, the face images shows more unexpected changes without a mask. We found that using the first 8 vectors gave the best trade-off in allowing the emotion to transfer without adding too many artifacts.

4.3 Learning Facial Expression Directions with SVM

Besides using logistic regression as the classifier to train our emotion directions, we also tried SVMs with different kernel functions to predict non-linear emotion directions. We used a linear activation layer as the last output layer in the SVM model. Figure 5 and 6 shows decent results with the “random_uniform” kernel function. However, none of our tests performed significantly better visually than our logistic regression model, so we did not pursue them when doing our experiments on our full model.

4.4 Scripting Emotion Videos

Our video generation model generates short face related videos from a single starting image. It generates from the starting image a sequence of plausible next emotions/face pose changes and then uses these together with the starting image and

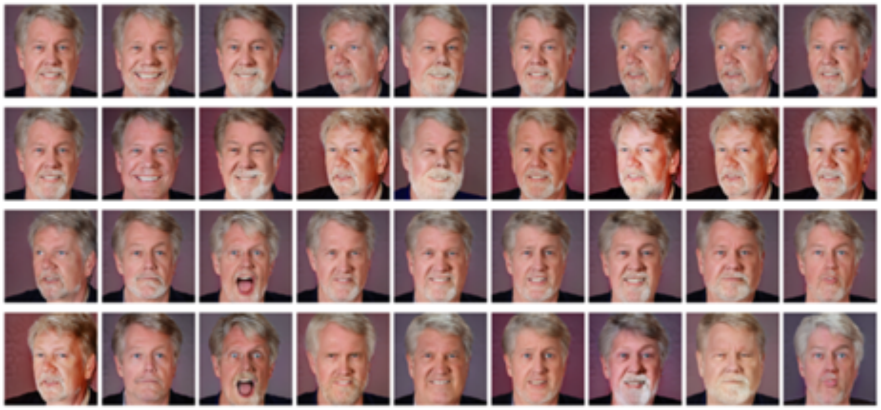


Figure 4: Generated images with mask and without mask. Line 1 and 3 are with a mask, while line 2 and 4 are without a mask.

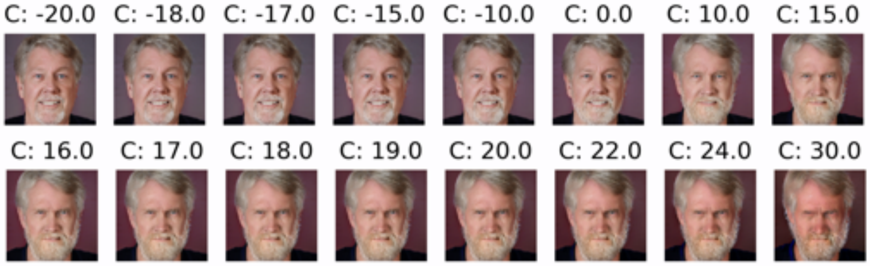


Figure 5: SVM Facial Expression Prediction - Anger

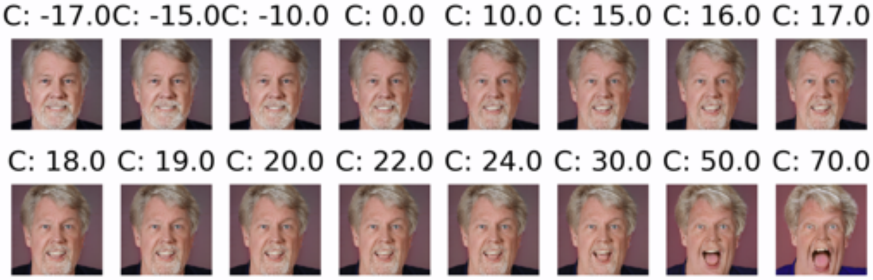
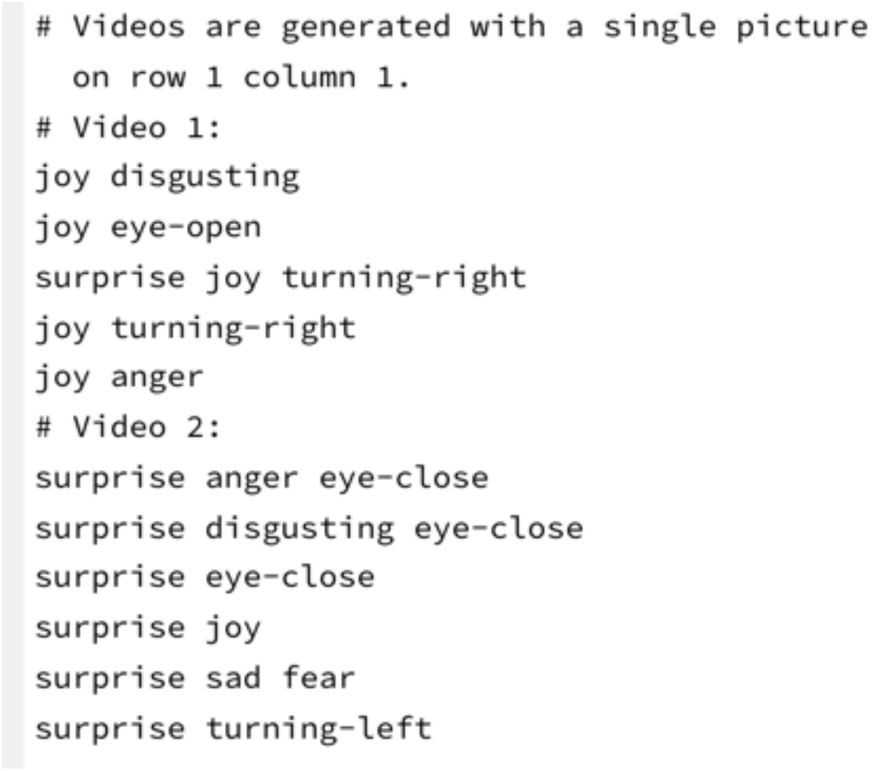


Figure 6: SVM Facial Expression Prediction - Surprise

trained feature directions for these emotions and pose changes to generate a video. As a potentially useful sub-process of this, one could imagine generating from a starting image and a user-scripted sequence of pose/emotion changes a video. We wrote a short program based on our model which takes as input an image file and a text file with pose emotion information to test this idea. We trained both emotions and face pose together in the same manner without any difference. Figure 7 shows a sample text file that could be used with our program, and Figure 8 shows the generated keyframes based on this text. As can be seen from the input file, we allow more than one emotion to be simultaneously expressed. For example, we can see this in the line: "surprise anger eye-close." Using this text file approach with lines like this, made it easier to quickly conduct experiments where multiple feature directions were simultaneously added.

4.5 Results

To statistically visualize how well our video synthesis algorithms work in easily presentable figures, we generated six facial expressions at 1024×1024 resolution starting from three random generated faces. We additionally generated three frames



```
# Videos are generated with a single picture
  on row 1 column 1.
# Video 1:
joy disgusting
joy eye-open
surprise joy turning-right
joy turning-right
joy anger
# Video 2:
surprise anger eye-close
surprise disgusting eye-close
surprise eye-close
surprise joy
surprise sad fear
surprise turning-left
```

Figure 7: Text human facial expressions.

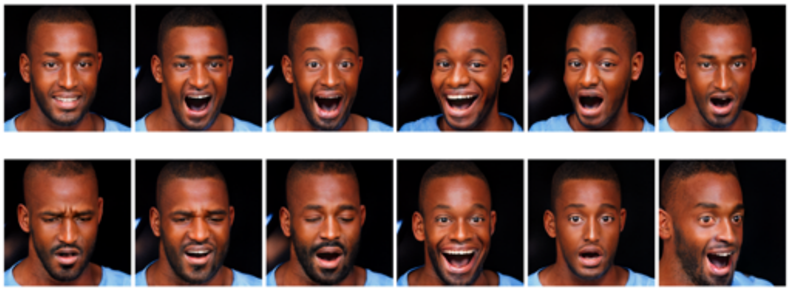


Figure 8: Generate keyframes with text. The top left image is the original start image and all other images are generated with text.

for each of the six expressions. In Figure 9, Figure 10, and Figure 11, we show the results of this visualization.

We do not use the Fréchet Inception Distance (FID) score to measure image diversity because we are limited to StyleGAN's output and its ability to generate new images. Since we are using StyleGAN we focused on making the best generated videos from a starting person and plausible changes that preserve that person's identity and that can be computed within the StyleGAN latent space. We used the Average Content Distance (ACD) [11] metric to measure content consistency of a generated video. The ACD is calculated using the average L_2 distance among all consecutive frames in a video. A smaller ACD score is better and means a generated video is more likely to be of the same person. We generated 210 videos using 35 randomly generated faces with each face having the same 6 different facial expressions. As shown in Table 1, our model shows the best results in generating consistent facial expressions as compared to TGAN and MoCoGAN.

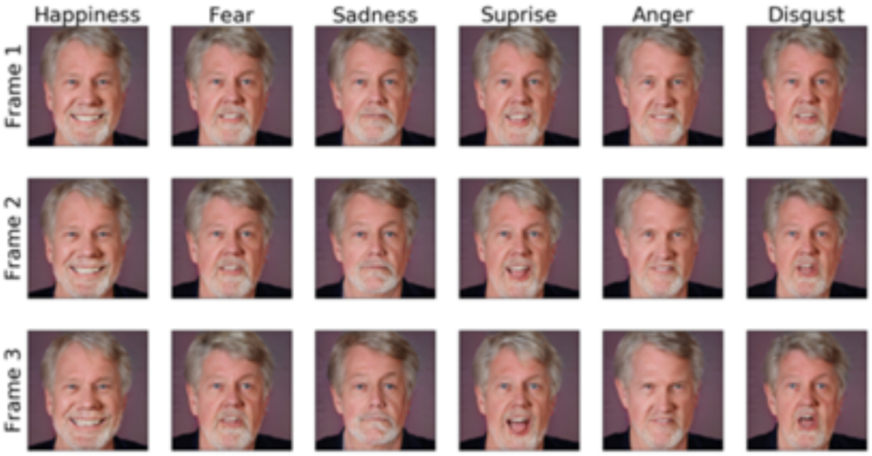


Figure 9: Facial Expressions with IMPA-FACE3D Database.

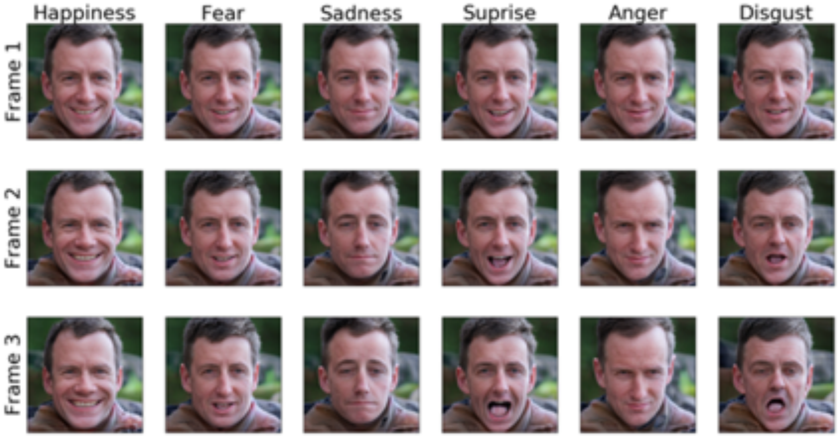


Figure 10: Facial Expressions with IMPA-FACE3D Database.

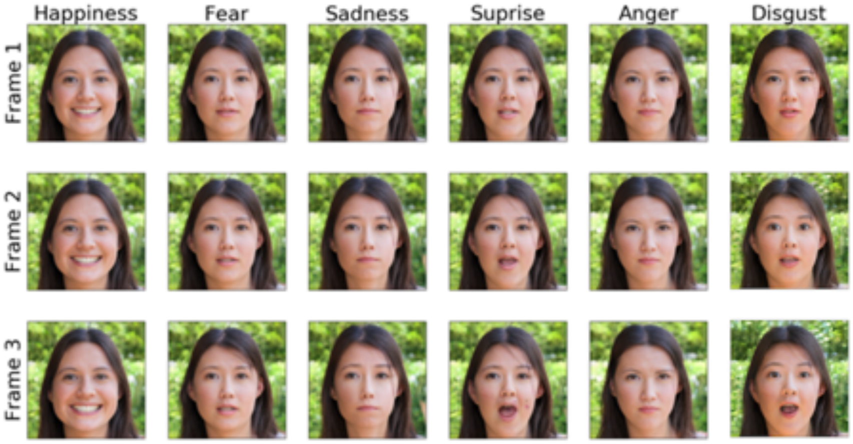


Figure 11: Facial Expressions with IMPA-FACE3D Database.

5 Conclusion

Motivated by the limitations of GANs to directly generate videos, we proposed a new method to synthesize videos from a pre-trained image GAN latent space. For our experiments with our method, we used the StyleGAN latent space because it

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

Table 1: Video generation content consistency comparison.

can produce good quality upscaled images, however, our method can be used with the latent space of any image GAN. Our results show that our method not only improves the speed of video generation as compared to prior work on transfer learning from image GANs, but also that it is suitable for generating high-resolution video clips. To the best of our knowledge, there are currently no direct-video GAN systems capable of generating video clips with 1024×1024 resolution other than ours. Furthermore, we feel our system is quite flexible. Using a pre-trained 2D image latent space with well-selected video frames, we expect to be able to predict better videos in the future. In addition, our approach also allowed one to mix several facial features at once simply by adding two or more emotion directions while delivering decent results. We see two potential directions for continuing our research: It should be possible to predict the directions in the latent space from self-labeled data, so it would be nice to carry out experiments in this direction. It should also be possible to modify our techniques to do random video synthesis assuming a proper loss function. One limitation of our results is that it relies on aligned images and we found it difficult to properly recover the frames of a video without image alignment. Thus, it would be nice to develop techniques that avoid image alignment to increase the class of videos that can be generated from our basic approach.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Perez Angelica. Emopy: A machine learning toolkit for emotional expression, 2018.

[3] Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[7] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019.

[8] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017.

[9] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[10] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2830–2839, 2017.

[11] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[12] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016.

[13] Yeyao Zhang, Eleftheria Tsipidi, Sasha Schriber, Mubbasir Kapadia, Markus Gross, and Ashutosh Modi. Generating animations from screenplays. *arXiv preprint arXiv:1904.05440*, 2019.