Infinite Expressions : GAN-Driven Facial Expression Augmentation

Aniket Ladukar
University of Southern California
ladukar@usc.edu

Kunal Bhujbal
University of Southern California
kbhujbal@usc.edu

Namrata Sharma
University of Southern California
nsharma4@usc.edu

Rajdeep Mahal
University of Southern California
rmahal@usc.edu

Sreya Reddy Chinthala
University of Southern California
chinthal@usc.edu

Tianrui Xia
University of Southern California
tianruix@usc.edu

Abstract

This project aims to revolutionize the integration of motion and expression into static images through the development of a Generative Adversarial Network (GAN) driven framework. By leveraging the First Order Motion Model, our approach seamlessly transfers dynamic facial expressions and head movements from high-quality video datasets onto corresponding still images. The core innovation lies in the synergistic collaboration of multiple deep learning models, including a keypoint detector, dense motion network, and an occlusion-aware generator. The keypoint detector maps facial features across input frames, while the dense motion network predicts dense motion fields that align source and driving poses. The generator warps the source image according to these motion fields, inpainting occluded regions to render photorealistic animated outputs. Extensive experiments on the VoxCeleb dataset validate our model's capability to breathe life into static portraits, paving the way for revolutionizing storytelling in entertainment, education, and historical preservation. Our framework's self-supervised training strategy circumvents the need for heavily annotated data, enabling generalization across diverse object categories. Additionally, we explore techniques to enhance training stability, including perceptual loss, equivariance constraints, and adversarial discriminator feedback. The results demonstrate our model's prowess in animating facial imagery with unparalleled realism and fidelity, unlocking new frontiers in immersive multimedia experiences.

1 Introduction

1.1 Motivation

Traditional static images often fail to capture the depth and vitality of real-life expressions and movements. Through this project, our aim is to bridge this gap and imbue still photographs with a sense of life and motion.

An important use case can be revolutionizing the entertainment industry by enabling the creation of lifelike animated characters. Our project seeks to develop a way for static images to be animated with

dynamic facial expressions and head movements, resulting in captivating characters for use in films, television shows, and video games. This innovative approach has the potential to elevate storytelling and immerse audiences in narratives like never before.

Moreover, beyond the realm of entertainment, our project holds significant implications for education and historical preservation. Imagine the possibility of animating historical photographs, thereby bringing iconic figures from the past to life. Such an advancement could revolutionize the teaching of history, making it more engaging and relatable for students.

1.2 Problem Statement

The primary challenge lies in seamlessly integrating dynamic facial expressions and head movements from high-quality video datasets onto corresponding still images. **We aim to extract motion of a subject from their facial video i.e** *a driver video* **and transfer them on to a still face image i.e** *source image*.

Figure 1: Example of source and driver images combined to get a combined animation

2 Related Work and Background

[2] The paper "Image-to-Image Translation with Conditional Adversarial Networks" introduces a novel approach to image-to-image translation utilizing conditional adversarial networks (cGANs). Image-to-image translation involves the transformation of an image from one representation to another, encompassing tasks such as converting label maps to photographs, sketches to photographs, or maps to aerial imagery. The proposed generator architecture adopts a "U-Net" design, facilitating the integration of low-level information across the network. In contrast, the discriminator is designed to penalize structure discrepancies at the scale of image patches, thereby enforcing local realism. Additionally, an L1 loss term is incorporated to ensure global consistency with the ground truth output. Through meticulous ablation studies conducted on the Cityscapes dataset, the paper meticulously examines the contributions of various components of the loss function and assesses the impact of architectural choices.

The key contribution of [3] is an innovative approach to unsupervised image translation leveraging generative adversarial networks (GANs) and cycle-consistency loss. The methodology entails training two generators (G and F) to translate between image domains X and Y bidirectionally. Adversarial loss functions are employed to ensure that the translated images closely resemble real images in the target domain. Additionally, cycle-consistency loss terms are introduced to enforce the condition that mapping an image from X to Y and back should faithfully reproduce the original image. This mechanism effectively prevents arbitrary mappings unrelated to the image content. The proposed CycleGAN method demonstrates remarkable efficacy across a diverse array of tasks, including collection style transfer, object transfiguration, season transfer, and photo enhancement.

Karras et al. present a seminal contribution to generative modeling with their paper, "Progressive Growing of GANs for Improved Quality, Stability, and Variation" [4] . Their innovative approach addresses inherent challenges in training Generative Adversarial Networks (GANs) by gradually increasing image resolution and model complexity during training. This progressive training strategy mitigates issues such as mode collapse and training instability, resulting in improved convergence and higher-quality generated images. The authors elucidate key techniques employed, including incremental growth of generator and discriminator networks, as well as the utilization of minibatch standard deviation to enhance sample diversity. This paper demonstrates significant enhancements in image quality, stability, and variation compared to conventional GAN training methods.

Another notable contribution to the realm of generative modeling is the paper "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network" [5]. Ledig et al. propose a novel methodology for single image super-resolution by leveraging Generative Adversarial Networks (GANs). Their approach is centered on the integration of perceptual and adversarial loss functions, enabling the generation of high-quality, photo-realistic super-resolved images from low-resolution inputs. The architecture combines the strength of both discriminative and generative networks, with the discriminative component ensuring the realism of generated images while the generative component focuses on maximizing perceptual similarity to ground truth high-resolution images. This paper demonstrates superior performance compared to traditional interpolation-based methods and state-of-the-art super-resolution techniques.

Additionally, Siarohin et al. propose a model architecture that can be used to animate any particular still image i.e *source image* based on motions in another video i.e *driver video*. It highlights the shortcomings of commonly used Deep Generative models used to transfer motion, stemming from their reliance on pre-trained object specific models which are built using heavily annotated source data, something that is usually not available for a random object category. Their paper [7] draws inspiration from Monkey-Net which is an object-agnostic model. It learns to encode motion information in the form of keypoints in a self supervised manner and uses this to predict keypoint transformation on the source image for each frame of the driving video. This eliminates the need to have object specific data annotations and allows the model to generalize well over an unseen set of objects. However, Monkey Net struggles to accurately model large object pose changes. This is due to its simplistic zeroth order modeling of object pose transformations. The paper proposes a first order model with two key improvements: introduction of regional affine tranformations applied around the keypoints and implementing a occlusion aware generator to mitigate the occlusion effects in the driving video. The complex modeling of motion transformations around keypoints along with the added robustness provided by the occlusion aware generator, allows this first order model to outperform all of its predecessors and generalize exceptionally well over unseen objects even in cases of significant pose changes.

3 Data

3.1 Description

[1] VoxCeleb is a large-scale speaker identification dataset created by researchers at the University of Oxford, one of the largest publicly available datasets for speaker identification research. It contains over 1 million utterances for over 7,000 celebrities, extracted from YouTube videos. This dataset is used for training the model.

There are several reasons why we chose this dataset for this project.

1. **Large-scale dataset:** VoxCeleb contains a large number of audio-visual recordings, featuring celebrities from various domains such as movies, television, and music. This large-scale dataset provides ample data for training complex models like GANs, which require significant amounts of data to learn and generate realistic outputs.
2. **Diverse set of speakers:** The dataset includes recordings of speakers from diverse backgrounds, representing different genders, ages, ethnicities, and accents. This diversity ensures that the trained models can generalize well and produce outputs that are representative of a wide range of human voices and appearances.
3. **High-quality recordings:** VoxCeleb contains high-quality audio-visual recordings, typically sourced from publicly available videos and interviews featuring celebrities. These recordings

often have high resolution and clear audio, which is crucial for training GANs to generate realistic audio-visual content.

4. **Pre-processed annotations:** VoxCeleb comes with pre-processed annotations, such as speaker identities, timestamps, and metadata. These annotations provide valuable information that can be used for supervised or semi-supervised learning tasks, as well as for evaluating the performance of trained models.
5. **Community support and benchmarking:** Due to its popularity, VoxCeleb has garnered significant attention from the research community. As a result, there are numerous pre-trained models, benchmarking studies, and research papers that leverage VoxCeleb for various tasks.

The source facial images that are animated with movements and emotions also come from the VoxCeleb dataset.

3.2 Preprocessing

One challenge encountered while working with the VoxCeleb dataset is the presence of encrypted, corrupted, or empty-frame videos. These videos pose a hindrance to the training process and need to be excluded from the training set. Addressing this issue involves several steps:

1. **Initial Integrity Check:** Utilizing ffprobe, a multimedia stream analyzer, we assess the integrity of each video file. If ffprobe returns a non-zero exit code, indicating an error, the video file is flagged as corrupt.
2. **Secondary Validation:** Even if ffprobe deems a file as valid, it may still be considered corrupted if it contains only identical frames. To ascertain this, we meticulously analyze each frame of the video. By computing the Mean Squared Error (MSE) between successive frames, we determine whether any substantial differences exist. If the MSE surpasses zero, indicative of variance between frames, the video is considered valid. Otherwise, it is marked as corrupted and subsequently excluded from the dataset.

4 Task and Approach

We utilized the First Order Motion Model for the task at hand [7]. This model necessitates two inputs: the source image, denoted as S, and a driving video, referred to as D. It's imperative that the object depicted in S and the object in motion within D belong to the same category or class. Subsequently, the model generates a video, which is essentially a sequence of image frames, depicting the object in S executing the same motion as observed in D. In the realm of computer vision, this problem is commonly referred to as Image Animation.

For training purposes, a repository of videos containing objects of the same category, as detailed in the data sources section, is required [1]. VoxCeleb is utilized for this. Since direct supervision is lacking (i.e., videos where objects exhibit similar motion), the First Order motion model adopts a self-supervised learning approach. During the training phase, each video is processed, and two frames are extracted from the same video clip. One frame is designated as the 'Source', while the other serves as the 'Driving' frame. Consequently, the Source and Driving frames feature the same object. However, during testing, a source image and a driving video are provided, and the model animates the object in the source image based on the motion exhibited by the object in the driving video.

4.1 Model Overview

The First Order Motion Model comprises multiple deep learning models, including the KP detector, Dense Motion network, Generator, and Discriminator. Among these, the Generator serves as the primary component responsible for predicting and generating the output, which corresponds to the driving frame. The subsequent sections will elucidate the processes involved in prediction by the Generator:

- Generator takes 2 inputs, source image and the output from the dense motion network (deformation and occlusion map), and produces the output.

- Dense motion network takes 3 inputs, source image, kp-source and kp-driving and produces the output.
- KP detector takes 2 inputs, source and driving image and outputs the keypoints predicted from both the inputs, kp-source and kp-driving.

In the reference paper [7], the author defines the first order motion model as a framework which takes inputs and produces an output. The framework has two main modules, motion estimation module and image generation module. The motion estimation module has 2 models, KP detector and Dense motion network. The image generation module has only a single model, which is the generator. The author ignores the discriminator in the framework actually.

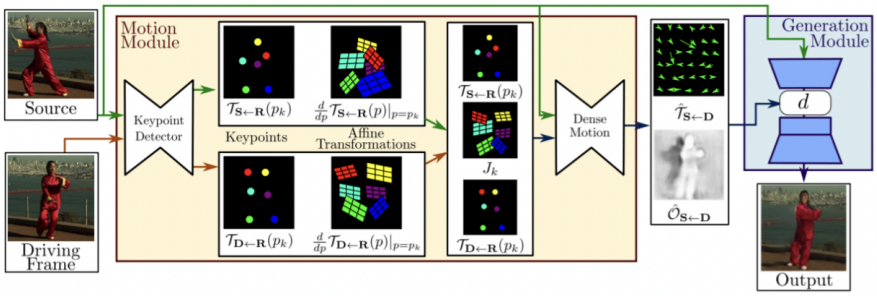


Figure 2: Model overview

4.2 Approach

1. The purpose of the motion estimation module is to predict a dense motion field from a frame $D \in R$ of the driving video D to the source frame $S \in R$.
2. The dense motion field is later used to align the feature maps computed from S with the object pose in D.
3. It is assumed that there exists an abstract reference frame R. Two transformations: from R to S (TS←R) and from R to D (TD←R) are estimated independently. The reference frame is an abstract concept that cancels out in derivations. This choice allows independent processing of D and S.
4. In addition, dense motion network outputs an occlusion mask O^S←D that indicates which image parts of D can be reconstructed by warping of the source image and which parts should be inpainted, i.e, inferred from the context.
5. Finally, the generation module renders an image of the source object moving as provided in the driving video. A generator network G warps the source image according to T^S←D and inpaints the image parts that are occluded in the source image.

4.3 Mathematics behind GAN

- **Error Function (Binary Cross-Entropy):** The binary cross-entropy error function, denoted as E, measures the difference between the true labels and the predicted probabilities by the discriminator. It is defined as:

$$E(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

  where:
  - y is the true label (0 or 1),
  - $\hat{y}$ is the predicted probability by the discriminator.
- **Value Function (GAN Objective):** The value function, denoted as $V(G, D)$, represents the objective function of the GAN. It measures the difference between the distribution of real data and the distribution of generated data. It is defined as:

$$V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}}[\log(D(x))] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]$$

  where:

- p_{data} is the distribution of real data,
- p_z is the distribution of noise input to the generator $G(z)$,
- $D(x)$ is the discriminator's output probability for real data x,
- $D(G(z))$ is the discriminator's output probability for generated data $G(z)$.

- **Optimal Discriminator:** The optimal discriminator D^* for a given generator G is the discriminator that maximizes the value function. It is defined as:

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

where:

- $p_g(x)$ is the distribution of generated data.

- **KL Divergence and JSD:**
  - **KL Divergence:** KL divergence measures the difference between two probability distributions P and Q. It quantifies how much one distribution diverges from another. For discrete distributions:

$$D_{KL}(P||Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right)$$

and for continuous distributions:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

  - **Jensen-Shannon Divergence (JSD):** Jensen-Shannon divergence is a symmetric and smoothed version of KL divergence, measuring the similarity between two distributions. It is defined as:

$$JSD(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M)$$

where $M = \frac{1}{2}(P + Q)$ is the average distribution.

- **Wasserstein Distance:** Wasserstein distance, also known as Earth Mover's Distance (EMD), measures the minimum amount of work required to transform one distribution into another. In the context of GANs, using Wasserstein distance as a metric can lead to more stable training and better convergence compared to traditional metrics like Jensen-Shannon divergence.

$$W(P, Q) = \inf_{\gamma \in \Gamma} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y)\, d\gamma(x, y)$$

where Γ is the set of all joint distributions with marginals P and Q, and $c(x, y)$ is the cost of transporting mass from x to y.

- **Training Stability:** GAN training can suffer from instability, which manifests as mode collapse or oscillation. Techniques such as gradient penalty and spectral normalization have been proposed to improve stability [6]. Gradient penalty introduces a penalty term to the loss function to enforce the Lipschitz constraint on the discriminator, while spectral normalization normalizes the spectral norm of weight matrices to stabilize the discriminator's output.

- **Gradient Penalty:** Gradient penalty is a regularization technique used to enforce the Lipschitz constraint on the discriminator. [6] It adds a penalty term to the loss function that penalizes the gradients of the discriminator's output with respect to its inputs. The penalty term encourages smoothness in the discriminator's decision boundary and improves training stability.

$$\text{Penalty} = \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right]$$

where $\hat{x}$ is a random sample interpolated between real and generated data, and λ is a hyperparameter controlling the strength of the penalty.

- **Spectral Normalization:** Spectral normalization is a technique used to stabilize the training of the discriminator by normalizing the spectral norm of weight matrices. [6] It constrains the Lipschitz constant of the discriminator's layers, making the discriminator's output more consistent and improving training stability.

$$W = \frac{W}{\sigma}$$

where W is the weight matrix and σ is its spectral norm.

.

4.4 Loss functions

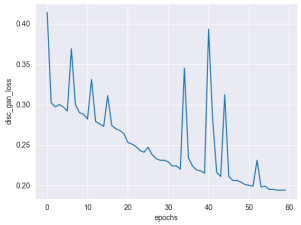
The loss function stands as the third most crucial element following data and model selection. Establishing the loss function simplifies subsequent steps involving backpropagation and gradient descent utilizing PyTorch. Similar to various models, multiple loss functions are employed to train each model effectively. These encompass perceptual loss, utilized for training both the generator and dense motion network, equivariance loss employed for training the KP detector, and discriminator loss, integral for training the discriminator. Figure 3 shows the values of losses during the process of training of the GAN.

1. Perceptual Loss: The primary loss function driving the first-order motion model, this perceptual loss function takes the original driving frame D and the predicted D' (generated by the generator) as inputs, yielding a loss value. This value is instrumental in updating both the generator and the dense motion network.

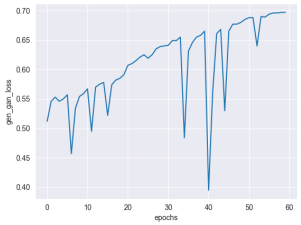
   The rationale behind employing perceptual loss lies in the necessity for D and D' to be perceptually similar. While simple mean squared error (MSE) loss could quantify the direct distance between the two, it falls short in capturing perceptual similarity. For instance, introducing Gaussian noise to D may render it visually identical to humans, yet MSE loss would regard them as dissimilar. Instead of directly measuring the distance between D and D', perceptual loss calculates the distance between multiple features extracted from both images using a pretrained model. In the referenced paper, the author utilized VGG-19, extracting features from all layers of the VGG-19 feature extractor. The loss is then computed by averaging over the distances of multiple features. This constitutes the conventional approach to computing perceptual loss.

   We further enhances this perceptual loss by computing it multiple times across various resolutions of D and D', averaging the results. This process culminates in obtaining the final loss value, which serves as the optimization objective.
2. Equivariance loss: Equivariance loss is employed to update the keypoint (KP) detector, especially in scenarios where keypoints are not readily available in the dataset for training the KP detector. Consequently, the KP detector is trained in an unsupervised manner. The procedure operates as follows: given an image X, a transformation is applied to yield a new image, X'. This transformation typically involves two types: affine transformation and thin plate spline transformation. Subsequently, both images X and X' are passed through the KP detector to obtain their respective keypoints. The keypoints obtained from X' undergo a reverse transformation. Ideally, the keypoints of X and the transformed keypoints of X' should align perfectly. To enforce this alignment, a loss function is devised to measure the distance between the keypoints of X and the transformed keypoints of X'. In addition to imposing the equivariance constraint on the keypoints, the author extends this constraint to the Jacobians, further enhancing the robustness of the model.
3. Discriminator loss: This loss function serves to update the discriminator model, which operates akin to a classifier. The primary task of the discriminator is to classify the real data D as genuine (assigned a label of 1) and the generated data D' as counterfeit (assigned a label of 0). Consequently, the loss function comprises two terms: one evaluates the accuracy of the discriminator's predictions on D, while the other assesses the accuracy on D'. In this context, the accuracy metric is quantified by the mean squared error (MSE) loss between the predicted values and the ground truth values. Thus, the final form of the loss function can be expressed as follows:

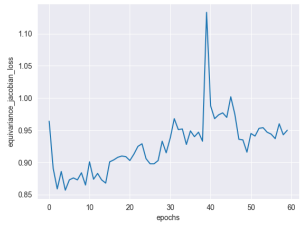
$$Discriminator\_loss = (1 - discriminator(D)) * *2 + discriminator(D') * *2$$



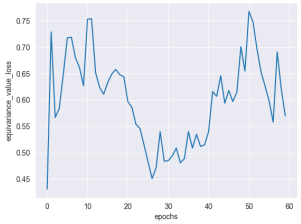
(a) Discriminator Loss



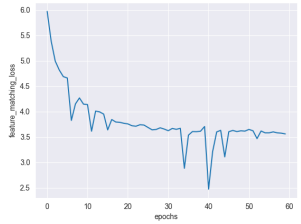
(b) Generator Loss



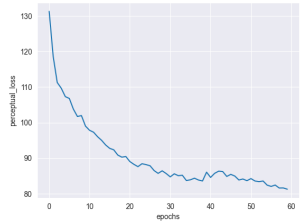
(c) Equivariance Jacobian Loss



(d) Equivariance Value Loss



(e) Feature Matching Loss



(f) Perceptual Loss

Figure 3: Various losses during GAN training

5 Results

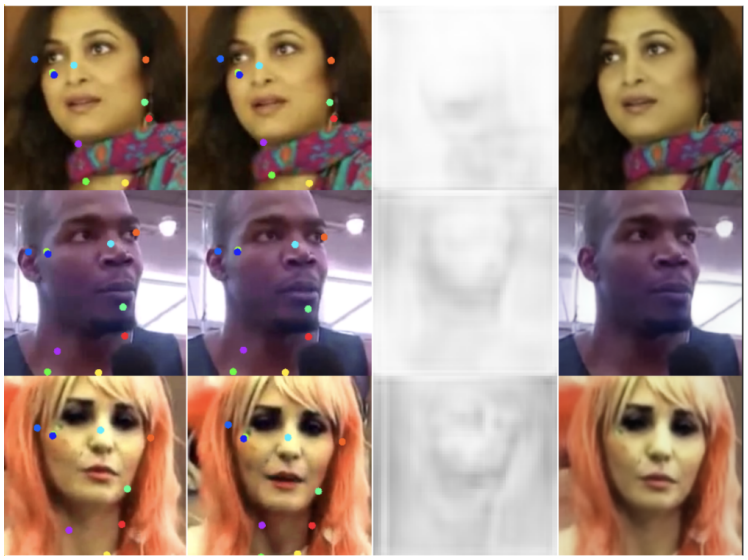


Figure 4: sample results

As shown in Figure 4: for each person, the first column represents the source image, which is the original pose or state of the object. The second column shows the driving image, which is the target motion or pose that the source image should mimic. The third column is the occlusion map, a binary or gray scale map that indicates the areas in the driving image that cannot be recreated from the source image by direct warping and thus need to be inpainted. Darker areas typically indicate parts that are occluded or not visible in the source image. The fourth column is the predicted output, which is the animation of the source image transformed to take on the pose or motion of the driving image.

In order for a more quantitative evaluation, we used various kinds of loss as performance metrics. As shown in Figure 4:

a&b. Discriminator Loss & Generator Loss: The plots for discriminator and generator losses reveal the adversarial characteristic of GAN training. The discriminator loss, which reflects its ability to distinguish real from fake, generally trends downward, suggesting that it is becoming better at its job over time. In contrast, the generator loss shows an overall upward trend, implying that it is facing increasing difficulty in fooling the discriminator. However, the spikes indicate periods where the generator produces better fakes that temporarily deceive the discriminator.

c. Equivariance Jacobian Loss: Measures how input transformations reflect in the output. The decreasing trend suggests improving equivariance, despite some training instability.

d. Equivariance Value Loss: Similar to the Jacobian loss, it gauges the network's consistency in handling input transformations. The significant reduction over time implies enhanced performance.

e. Feature Matching Loss: Aids in stabilizing training by comparing generated images to real ones at the feature level. A declining trend indicates increasing similarity between generated and real images.

f. Perceptual Loss: Measures the perceptual resemblance of generated images to real ones. The downward trajectory is encouraging, signaling growing realism in generated images.

In summary, the GAN is showing signs of learning with overall downward trends in most loss functions, but there are spikes and irregularities, which are not uncommon in GAN training. These spikes can result from the adversarial nature of the training process, where the generator and discriminator are constantly trying to outperform each other, leading to a dynamic training landscape.

6 Conclusion

Figure 5: Image 1: Source image, Image 2: Driver Video, Image 3: Augmented video

The synergistic combination of deep learning models, including the keypoint detector, dense motion network, and occlusion-aware generator, has proven instrumental in achieving photorealistic animation results. The keypoint detector's ability to map facial features across input frames, coupled with the dense motion network's prediction of dense motion fields, facilitates accurate alignment of source and driving poses. The generator's warping capabilities and inpainting of occluded regions further contribute to the realism and fidelity of the animated outputs.

Extensive experiments conducted on the VoxCeleb dataset have demonstrated the efficacy of our approach, validating its potential to revolutionize storytelling in the entertainment industry, educational spheres, and historical preservation efforts. By breathing life into static portraits, our framework offers a compelling solution for creating captivating animated characters, immersive learning experiences, and vivid historical recreations.

Moreover, the self-supervised training strategy employed by our model circumvents the need for heavily annotated data, enabling generalization across diverse object categories. This versatility opens up exciting avenues for future exploration, including domain generalization and the development of unified models capable of accommodating multiple applications.

In summary, our GAN-driven framework for facial expression augmentation represents a significant stride towards bridging the gap between static imagery and dynamic realism. By unlocking the power of motion and expression in still portraits, we have paved the way for immersive multimedia experiences that captivate audiences, educate in innovative ways, and preserve historical narratives with unprecedented vividness.

6.1 Challenges

There were a number of challenges we faced in the execution of the project:

- Model training requires huge amounts of data, often tens or hundreds of Gigabytes of data. A lot of computational resources are required to handle these amounts of data to produce a decent result.
- Due to resource constraints, the developed model fails to capture the intricacies of human expressions and the generated output lacks the details and the expressiveness of a human face.
- It is hard to design evaluation metrics for such a task. Using pixel based similarity metrics do not work due to the nature of the problem since they do not capture perceptual quality. We need to heavily rely on quality based metrics which depend on human perception, a highly subjective phenomenon. It is also very time consuming to gather human feedback. All these factors combined make designing the evaluation metrics a difficult tasks.
- There are cases where the model produces a very lifelike animations but it may still fail to appear humanlike. This is a common issue in animating human faces.

6.2 Future Work

There are a number of ways our work can be expanded:

- Scaling up the model and training process to leverage larger datasets and more computational resources, potentially leading to improved results and capturing finer details.
- Extending the model to handle more diverse object categories beyond human faces, leveraging the self-supervised training strategy and the model's potential for generalization, like motion of vehicles, animals, and even motion of entire human body.
- Exploring techniques to enhance training stability and convergence, as GAN training is known to be challenging and prone to instabilities, as evidenced by the spikes in the loss curves.
- Exploring techniques to improve the model's ability to generate truly human-like animations, even when the generated output appears lifelike, as mentioned in the challenges section.

References

[1] A. Zisserman A. Nagrani*, J. S. Chung*. Voxceleb: a large-scale speaker identification dataset, 2017.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[3] Phillip Isola Alexei A. Efros Jun-Yan Zhu, Taesung Park. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.

[4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017.

[5] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv:1606.03498 [cs.LG]*, 2016.

[7] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation, 2020.