Analysis and Modification of Encoded Single User Facial Expression Data

Colin Sullivan
Stanford University
colins26@stanford.edu

1 Introduction

Facial Expression Recognition (FER) is an exciting and now widely studied problem. Essentially, the aim is to be able to extract facial expressions from face image data, usually with the goal of interpreting the pictured person's emotional state. Though, the uses of such technology could certainly extend past such applications: FER could be used for robot companions to respond to their users' emotional states, for psychological diagnosis, and to monitor stress levels [1].

While this paper doesn't seek to reach state of the art performance for that specific task, it will take on a simpler version of the problem — compressing the facial image of a single subject. The goal of this study being to isolate as many variables as possible in order to analyze the extent to which pure facial image data can be compressed.

1.1 Motivation

While quite daunting for a large set of various persons' data, FER is much simpler under more uniform conditions (lighting, background, person, etc) [2]. Further, a lot of opportunity have recently arisen to collect and utilize individual users' facial data using mobile devices. Imagine the possibilities of learning ones own facial data distribution: one could make their own avatar for use in online communication, shift emotions while video conferencing to improve productivity and collaboration, improve facial recognition software for personal security, and much more.

We also wish to further our understanding of the nature of facial expression encoding. It might be useful to know, for example, how many latent variables are required to properly store a "facial expression" and which facial expressions are encoded similarly.

1.2 Related Work

As mentioned previously, FER itself is a widely studied field. There are several works on the topic. One such example being Usman's paper, "Using Deep Autoencoders for Facial Expression Recognition." This study isolated cropped facial images and attempted to compress this data as much as possible using deep stacked autoencoders and then used the compressed data to classify facial expressions [1]. This project will take a lot of inspiration from that one in terms of the goal of facial expression compression, but we will also be attempting to visualize and explore the generated latent space, similarly to what Yeh did in "Semantic Facial Expression Editing using Autoencoded Flow." This paper used a convolutional autoencoder to encode facial image data and then decoded the resulting latent space to a per-pixel flow field in order to get more detailed results translating facial images between various facial expressions [2]. There are a plethora of other works in the field of facial expression recognition, translation, and generation, including Ruiz, Boston University, and Apple's famous MorphGAN paper, "MorphGAN: One-Shot Face Synthesis GAN for Detecting Recognition Bias."

1.3 Goal

The goal of this project is to encode and analyze facial expression data of a single user with the aim of gleaning useful information from the distribution of data points in the latent space. For example, we would hope to discover some correlation or grouping of similar emotions in the image data (e.g. happy, sad, angry, etc). We also hope to use this information to interpolate between encoded facial expressions with the outcome hopefully being a shift in emotion of the pictured user.

2 Problem Statement

Given a half hour long video of a single user, the goal is to successfully learn an accurate distribution of their face on a compressed latent space.

2.1 Data

First, we will drop video frames to end up with an 8 fps video (about 14,000 frames). HAAR face detection will be run on each of the frames in this video to get square images centered on the user's face and these images will be cropped in order to avoid capturing hair, ears, or the background as much as possible. Finally, we will scale these images down to size 64 by 64.

2.2 Approach, Expected Results, and Evaluation

We will use a β-VAE to capture the distribution. After training is complete, we will sample three points in the latent space and interpolate between them to verify sample quality and check for mode collapse. Next, several more data points of smiling images will be processed and run through the encoder. We will test the grouping of these images and visualize their positions in the latent space to (hopefully) confirm the meaningfulness of the latent variables. After that, we will interpolate from other data points to the center of the smiling data points in order to, hopefully, get a nice facial expression shift. Lastly, we'll record a live demo of captured facial expressions mapping directly to a visualized point in the latent space to see how it moves around as the user's expressions change.

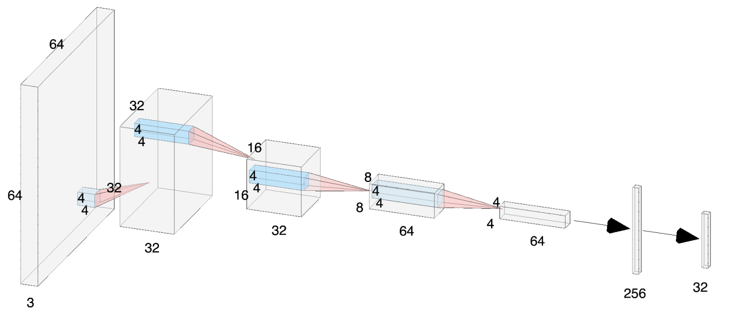
The primary metric of evaluation for this model will be the quality of its generated samples. We could also measure the disentanglement of the model's latent space and the PSNR of its image output if useful.

3 Technical Approach

We will use the architecture used to train a convolutional β-VAE on the CelebA datasert in "β-VAE: Learning Basic Visual Concepts With a Constrained Variational Framework":

Input: 64x64x3
Encoder: Conv (stride 2) 4x4x32, 4x4x32 4x4x64, 4x4x64, FC 256. ReLU activation.
Latents: 32
Decoder: Deconv reverse of encoder. ReLU activation. Gaussian.

Note that our latent will actually be split into two vectors of size 32: the μ vector and the σ vector. We will sample z from these vectors using the repameterization trick, $\epsilon = (0, I), z = \mu + \epsilon * \sigma$.



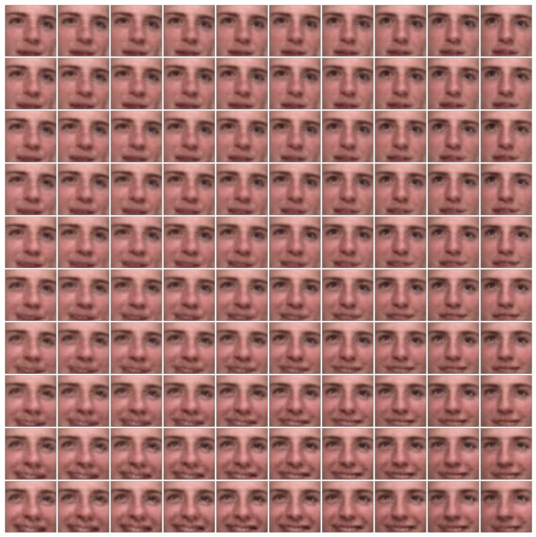
We will use the Adam optimizer with a 1e4 training rate and the usual lower bound β-VAE loss function:

$$\mathcal{L}(\theta, \phi; x, z, \beta) = \mathbb{E}_{q_\theta(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\theta(z|x)||p_\theta(z))$$

Where $p(z)$ is just the unit normal, $q(z|x)$ is our encoder and $p(x|z)$ is our decoder.

4 Preliminary Results

A good amount of work went into actually gathering the dataset, so there was not a lot of room for experimentation. That being said, I wanted to verify that the distribution of the gathered dataset was at least learnable, so I trained a convolutional Wasserstein GAN (leftover from another project) on the dataset overnight to learn the distribution and got some pretty nice (talking about the sample quality, not my face) interpolated images as a result:



While somewhat uniform as expected, we can see pretty clearly that smile size changes by the row and some combination of gaze direction and horizontal tilt changes by the column. These results definitely gave me hope that a model could actually generate some pretty good samples from this dataset, and I'm very excited to see the results of the upcoming experiments!

References

[1] Using Deep Autoencoders for Facial Expression Recognition (https://arxiv.org/pdf/1801.08329.pdf)

[2] Semantic Facial Expression Editing using Autoencoded Flow (https://arxiv.org/pdf/1611.09961.pdf)

[3] MorphGAN: One-Shot Face Synthesis GAN for Detecting Recognition Bias (https://arxiv.org/pdf/2012.05225.pdf)

[4] VAE Generative Modelling for Face Recognition (https://medium.com/analytics-vidhya/vae-generative-modelling-for-face-recognition-71e8ba16950c)

[5] β-VAE: Learning Basic Visual Concepts With a Constrained Variational Framework (https://openreview.net/pdf?id=Sy2fzU9gl)

[6] Autoencoder Downsampling and Upsampling (https://kharshit.github.io/blog/2019/02/15/autoencoder-downsampling-and-upsampling)