Time Travel using Conditional Adversarial Auto Encoder

DAAN 897-Deep Learning, April 2019

Anchal Gupta, Sindhu Thumati

Abstract

Time Travel is an application that aims to predict the progression and regression of a person’s face as a function of age. Given an image of a person at an arbitrary age, the application will predict how the person looked like in the past and how the person will look like in the future. Specifically, the output will consist of 10 predictions (as images) corresponding to 10 different age ranges (31-40 years, 41-50 years, etc.) without compromising the features of the person’s face.

Introduction

Face progression and regression has tremendous impact to a wide-range of applications like face prediction of missing/wanted people, entertainment, etc. But major challenge in this area of research come from the rigid requirement to the training and testing datasets, as well as the large variation presented in the face image in terms of expression, pose, resolution, illumination, and occlusion. In this paper, we investigate the age progression/regression problem from the perspective of generative modeling. We assume that the face images lie on a high-dimensional manifold and given a query face, we could find the corresponding point on the manifold. Stepping along the direction of age changing, we will obtain the face images of different ages while preserving personality. We propose a conditional adversarial autoencoder network to learn the face manifold. By controlling the age attribute, it will be flexible to achieve age progression and regression at the same time. Because it is difficult to directly manipulate on the high-dimensional manifold, the face is first mapped to a latent vector through a convolutional encoder, and then the vector is projected to the face manifold conditional on age through a deconvolutional generator. Two adversarial networks are imposed on the encoder and generator, respectively, forcing to generate more photo-realistic faces. The disentanglement of age and personality in the latent vector space helps preserving personality while avoiding the ghosting artifacts. We deviate from the popular group-based learning, thus not requiring paired samples in the training data or labeled face in the test data, making the proposed framework much more flexible and general.

Data Collection and Pre processing

The dataset for the neural network in this project is the UTKFace dataset [3] which contains 20,000 (200 x 200 px) aligned and cropped facial images of people of various race, gender and age.

Figure 1: Sample images from UTKFace dataset.

Some statistics about the dataset were determined and we found that it contains an almost equal number of males and females. On the other hand, there are more images of people below the age of 30 than people above the age of 70, as shown in Figure 2. The race distribution of the dataset is skewed towards the white race. Furthermore, some of the age labels were blatantly incorrect in which case we either removed or relabeled them. To reduce computational cost, the images were normalized and down-sampled to 128 x 128 px (resized and center-cropped). Then the images were labeled with the appropriate age ranges (one of 0-5, 6-10, 11-15, 16-20, 21-30, 31-40, 41-50, 51-60, 61-70, 70+) that were chosen to mitigate the in-balance of age in the dataset.

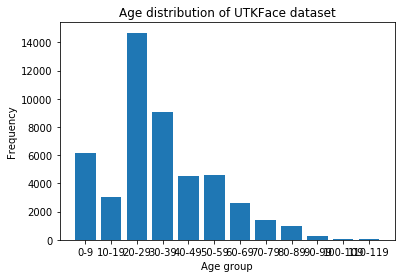


Figure 2: Age distribution of UTKFace dataset.

Software Architecture

Time Travel consists of a client and server side. On the client side, users will have the option to select an input image by browsing their computer. Once an image is submitted, a GET request is sent to the API back-end which calls use model.py to pre-process the input image. The new image is then passed to the pre-trained models to generate the 10 images that are displayed on the client side for users to observe. See Figure 3.

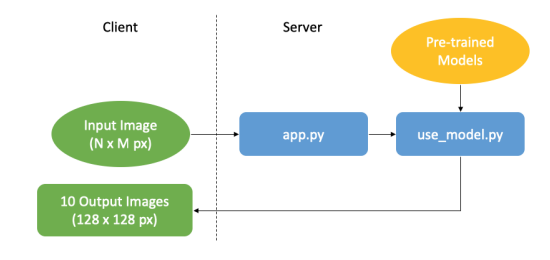


Figure 3: The structure of Time Travel’s software.

The Model: Conditional Adversarial Autoencoder

We propose a pipeline of conditional adversarial autoencoder (CAAE) network that learns the face manifold. The CAAE incorporates two discriminator networks, which are the key to generating more realistic faces. Below is the architecture of the CAAE implemented in this project.

* Conditional Adversarial Autoencoder

The input and output face images are 128 × 128 RGB images and x ∈ R 128×128×3 . A convolutional neural network is adopted as the encoder. The convolution of stride 2 is employed instead of pooling because strided convolution is fully differentiable and allows the network to learn its own spacial down sampling. The output of encoder E(x) = z preserves the high-level personal feature of the input face x. The output face conditioned on certain age can be expressed by G(z, l) = ˆx, where l denotes the one-hot age label. Unlike existing GAN-related works, we incorporate an encoder to avoid random sampling of z because we need to generate a face with specific personality which is incorporated in z. In addition, two discriminator networks are imposed on E and G, respectively. The Dz regularizes z to be uniform distributed, smoothing the age transformation. The Dimg forces G to generate photo-realistic and plausible faces for arbitrary z and l.

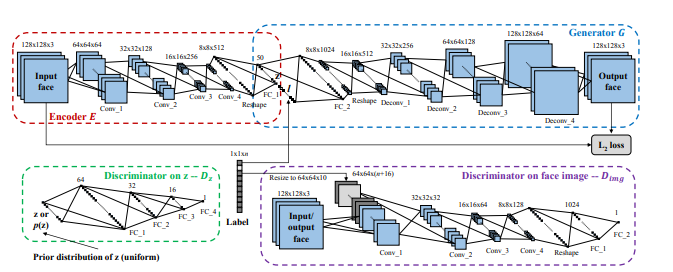


Figure 4: Architecture

We assume the face images lie on a high-dimensional manifold, on which traversing along certain direction could achieve age progression/regression while preserving the personality. However, modeling the high-dimensional manifold is complicated, and it is difficult to directly manipulate on the manifold. Therefore, we will learn a mapping between the manifold and a lower-dimensional space, referred to as the latent space, which is easier to manipulate. The real face images lie on the face manifold M, so the input face image x ∈ M. The encoder E maps the input face x to a feature vector, i.e., E(x) = z ∈ R n, where n is the dimension of the face feature. Given z and conditioned on certain age label l, the generator G generates the output face xˆ = G(z, l) = G(E(x), l). Our goal is to ensure the output face xˆ lies on the manifold while sharing the personality and age with the input face x (during training). Therefore, the input and output faces are expected to be similar as expressed in in the below equation (L(.,.) is the L2 norm)



* Encoder

The encoder is a convolutional neural network that receives a 128 x 128 px image as input and outputs a 50-element vector. It consists of 4 convolutional layers, 1 linear layer, and ReLU activations. A convolutional neural network is appropriate here because it is the architecture of choice for identifying features in images. Each convolutional layer has a kernel size of 5 x 5, stride of 2 (to reduce computational cost and allow the network to learn the appropriate down sampling of the image instead of manually pooling). A key insight is that the neural network is ’learning’ a high dimensional manifold of faces (each dimension represents a feature of the image) and the encoder is learning to represent this manifold with a space of simple 1-dimensional feature (latent) vectors. By keeping age independent of the other facial features, we can ’move’ through the latent space in the direction of increasing or decreasing age and project the vector back on the manifold 3 to recover the desired facial image. Mathematically, if we denote the manifold, M and an image x where x ∈ M, then z = E(x) represents the latent vector, and we can concatenate a one-hot age label, l, to get [z, l], a point in latent space. We can ’move’ in the latent space by changing l and generate a corresponding image y = G(z, l) conditioned on l. We would like to impose a uniform distribution on the encoder’s output so that space of feature vectors (representing personality) becomes populated evenly, therefore smoothing the aging process for people that the model has not trained on. We use the encoder as the input to our generator rather than random noise because we want the output to retain the identity of the person. We train E by maximizing the loss of Dz, given by log(Dz(E(x))).

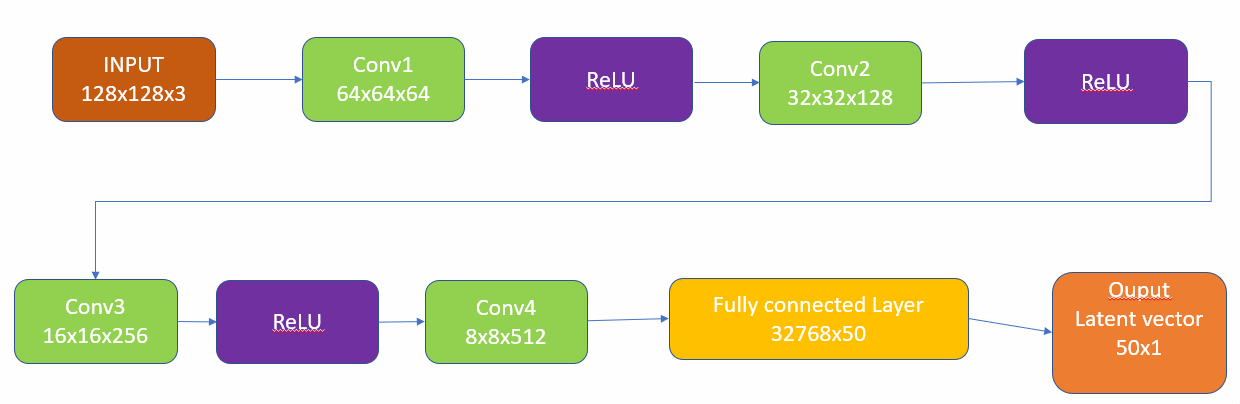


Figure 5: Architecture of Encoder

* Generator

The generator is a de-convolutional neural network that performs the operation of transforming a latent vector back into an image. The generator is learning a mapping between vectors in the latent space back to the manifold of faces. The architecture of the generator consists of 1 fully connected linear layer and 4 de-convolutional layers, with ReLU and Tanh (at the end) activations. Note that the pixels in the images and the latent vectors are normalized to [-1, 1] to improve training speed. The kernel size, number of input and output channels of the de-convolutional layers mirror the opposite process of the encoder. Ultimately, we want to be able to generate a person’s face conditioned on one of the 10 possible age ranges. The loss of the generator, G, can be expressed mathematically as a weighted average of kx − G(E(x), l)k and log(Dz(G(z))). The first term is the reconstruction loss between the input image and the generated image at the original age group and the second term represents the fact that we want to fool the discriminator into thinking the generated images are real.

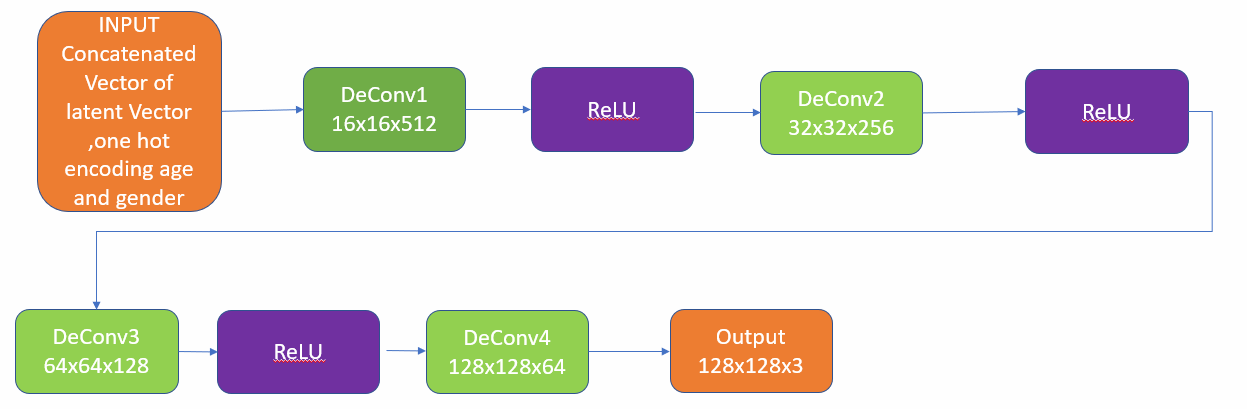


Figure 6: Architecture of Generator

* Discriminator on z

The discriminator on z, denoted by Dz, imposes a prior distribution on z. Specifically, Dz aims to discriminate the z generated by encoder E. Simultaneously, E will be trained to generate z that could fool Dz. Such adversarial process forces the distribution of the generated z to gradually approach the prior. We used uniform distribution as the prior, forcing z to evenly populate the latent space.



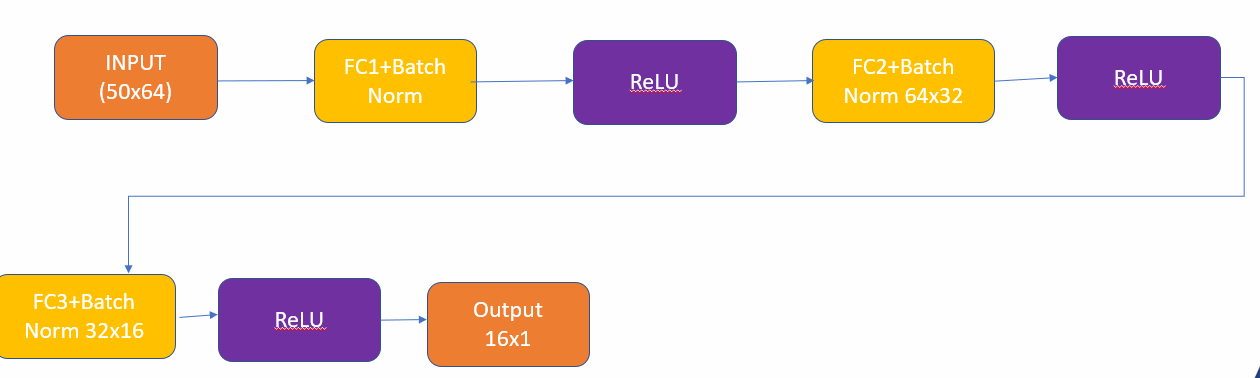


Figure 7: Architecture of Discriminator on Encoder

* Discriminator on face images

The discriminator Dimg on face images forces the generator to yield more realistic faces. In addition, the age label is imposed on Dimg to make it discriminative against unnatural faces conditional on age. Although minimizing the distance between the input and output images forces the output face to be close to the real ones, it does not ensure the framework to generate plausible faces from those unsampled faces. For example, given a face that is unseen during training and a random age label, the pixel-wise loss could only make the framework generate a face close to the trained ones in a manner of interpolation, causing the generated face to be very blurred. The Dimg will discriminate the generated faces from real ones in aspects of reality, age, resolution, etc. Dimg assists the framework to generate more realistic faces. The outputs without Dimg could also present aging but the effect is not as obvious as that with Dimg because Dimg enhances the texture especially for older faces. 

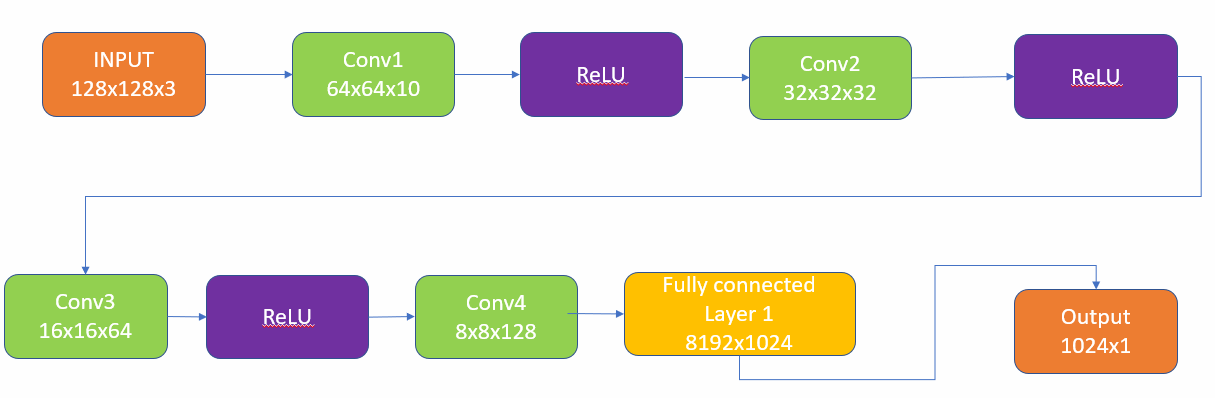
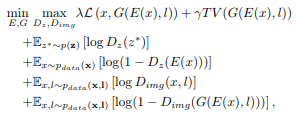


Figure 8: Architecture of Discriminator on face Image

Finally, the objective function of our network becomes



where T V (·) denotes the total variation, which is effective in removing the ghosting artifacts. The coefficients λ and γ balance the smoothness and high resolution. The age label is resized and concatenated to the first convolutional layer of Dimg to make it discriminative on both age and human face.

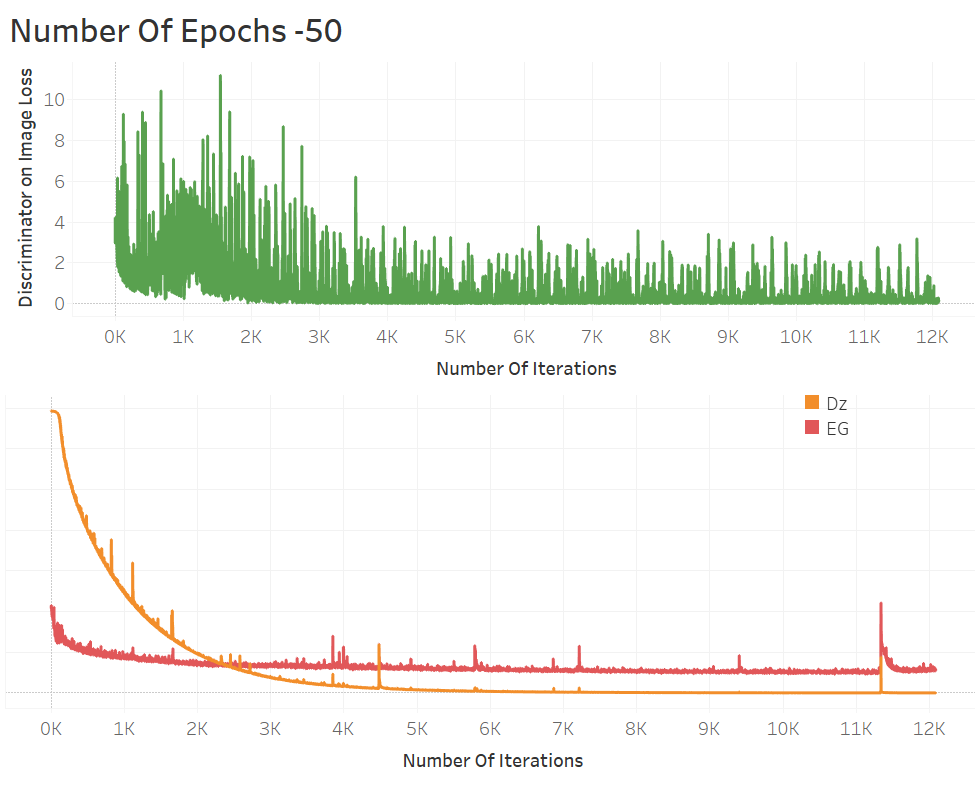
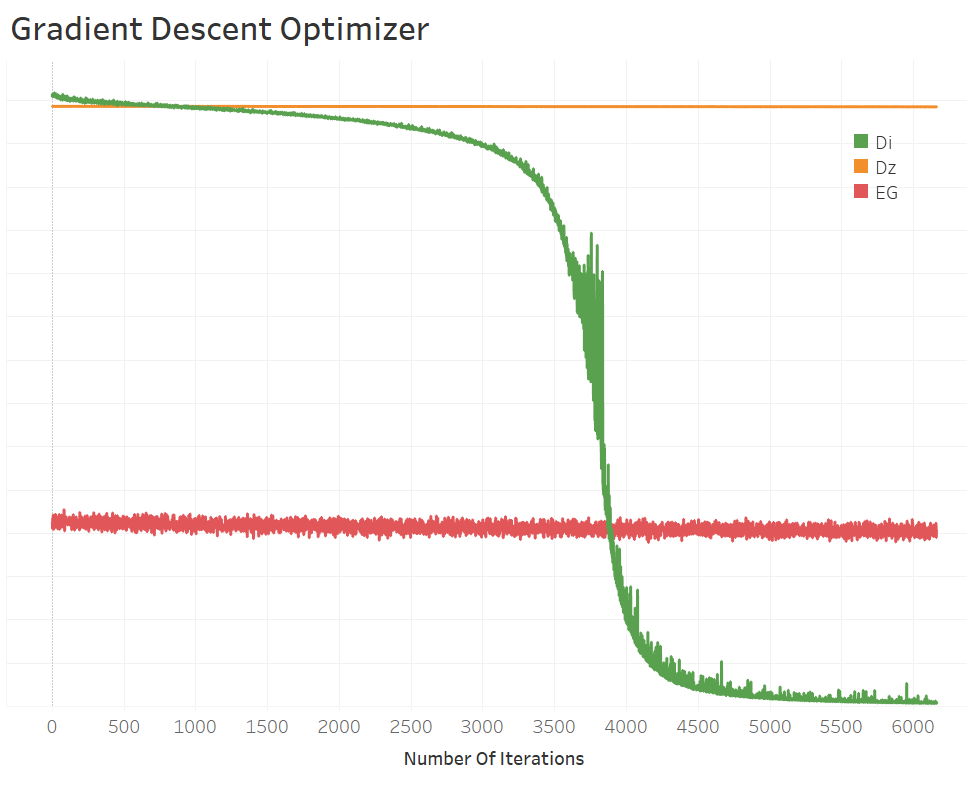
Implementation

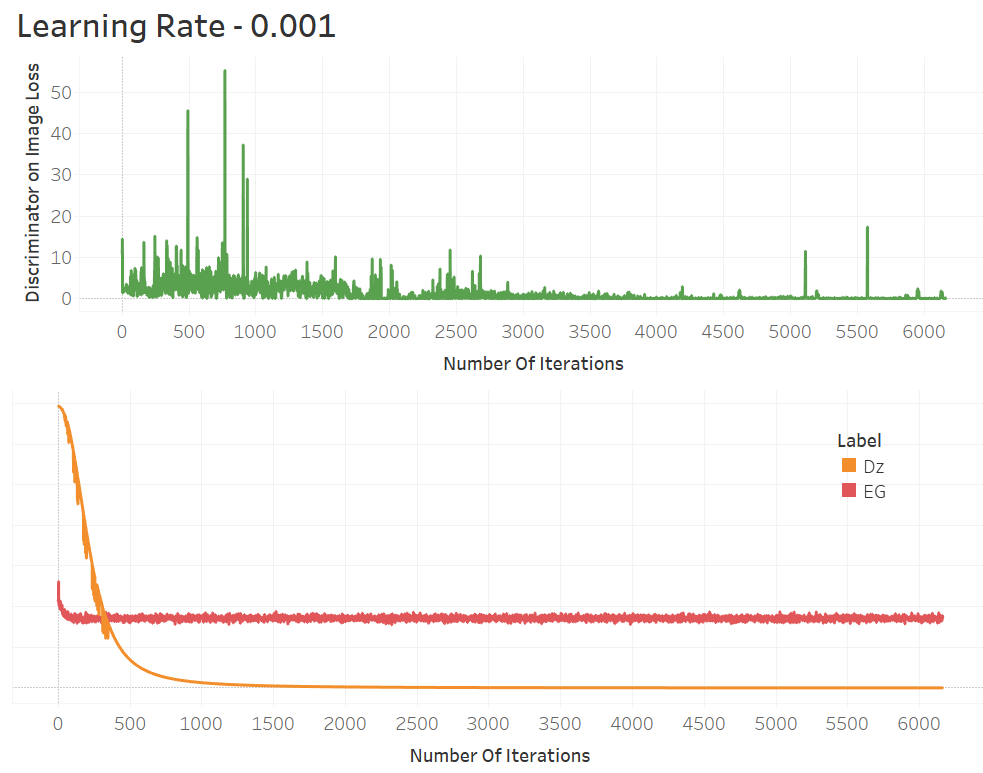
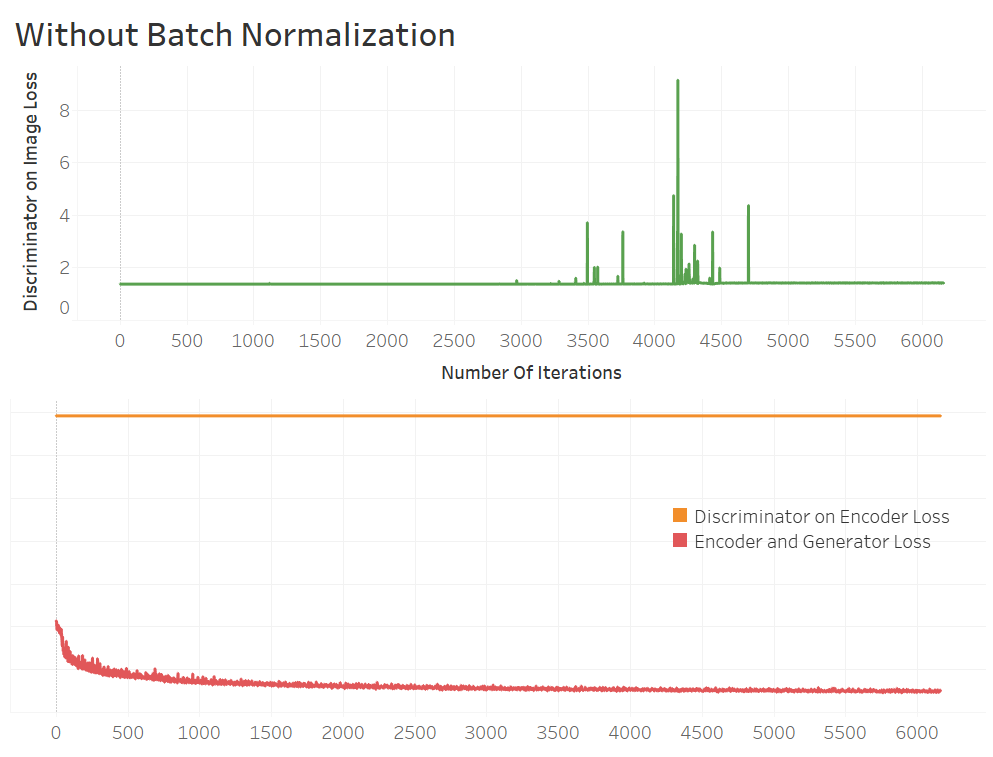
We built a network according to architecture illustrated in the above section with kernel size of 5 × 5. The pixel values of the input images are normalized to [−1, 1], and the output of E which is z is also restricted to [−1, 1] by the hyperbolic tangent activation function. Then, the desired age label, the one-hot vector, is concatenated to z, constructing the input of G. To make fair concatenation, the elements of label is also confined to [−1, 1], where -1 corresponds to 0. Finally, the output is also in range [−1, 1] through the hyperbolic tangent function. Normalizing the input may make the training process converge faster. We have not used the batch normalization for E and G because it blurs personal features and makes output faces drift far away from inputs in testing. However, the batch normalization will make the framework more stable if it is applied on Dimg. All intermediate layers of each block (i.e., E, G, Dz, and Dimg) use the ReLU activation function. In training, λ = 100, γ = 10, and the four blocks are updated alternatively with a mini-batch size of 100 through the stochastic gradient descent solver, ADAM [12] (α = 0.0002, β1 = 0.5). Face and age pairs are fed to the network. After about 50 epochs, plausible generated faces can be obtained. During testing, only E and G are active. Given an input face without true age label, E maps the image to z. Concatenating an arbitrary age label to z, G will generate a photo-realistic face corresponding to the age and personality.

Hyperparameter Tuning

We tried a variety of hyperparameter combinations to get the best model. Following figures compares the Dz, EG and Dimg loss.

We found that Adam quickly converged to reasonable qualitative results, so it was used instead of SGD. The losses for the various components of the neural network are shown in Figure 9. Most of the losses (E, G, Dimg, Dz) are small in magnitude and stay relatively constant. Notice that Gimg loss increases as the number of epochs increases. This indicates that Dimg is finding it difficult to guess real versus reconstructed images. The qualitative results of the training process (using a validation image) is shown in Figure 10, where age is increasing from left to right. The aging process is slightly evident even after 10 epochs, although the generated images are slightly blurry. On the other hand, the aging process after 50 epochs is much more evident and realistic.

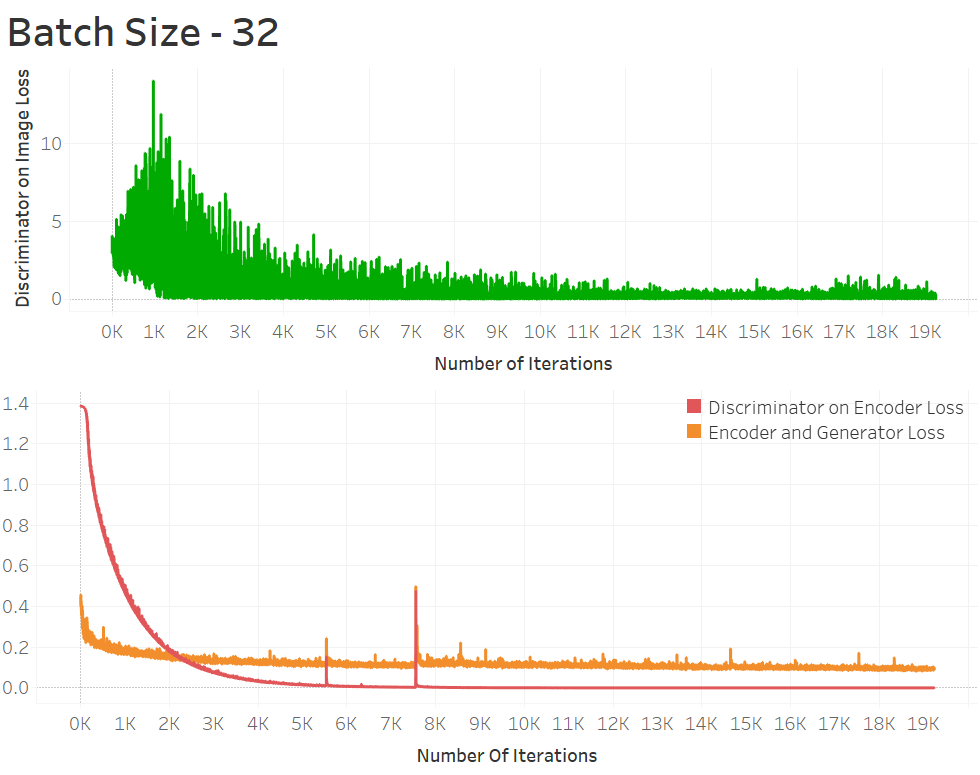


Figure 9: Loss for Various Hyperparameters

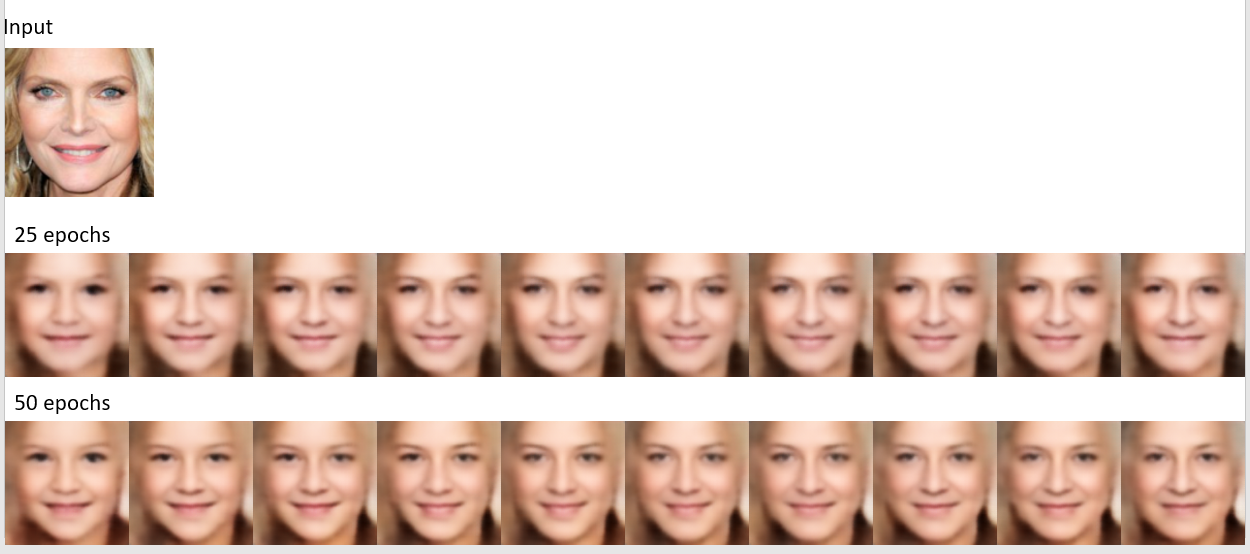


Figure 10: Sample output for image using model trained for 25 and 50 epochs.

Performance Comparison/ Discussion of Results

The proposed conditional adversarial autoencoder (CAAE), which first achieves face age progression and regression in a holistic framework. We deviated from the conventional routine of group-based training by learning a manifold, making the aging progression/regression more flexible and manipulatable — from an arbitrary query face without knowing its true age, we can freely produce faces at different ages, while at the same time preserving the personality. We demonstrated that with two discriminators imposed on the generator and encoder, respectively, the framework generates more photo-realistic faces. Flexibility, effectiveness, and robustness of CAAE have been demonstrated through extensive evaluation.

Future Work

The proposed framework has great potential to serve as a general framework for face-age related tasks. More specifically, we trained four sub-networks, i.e., E, G, Dz, and Dimg, but only E and G are utilized in the testing stage. The Dimg is trained conditional on age. Therefore, it is able to tell whether the given face corresponds to a certain age, which is exactly the task of age estimation. For the encoder E, it maps faces to a latent vector (face feature), which preserves the personality regardless of age. Therefore, E could be considered a candidate for cross-age recognition. The proposed framework could be easily applied to other image generation tasks, where the characteristics of the generated image can be controlled by the conditional label. In the future, we would extend current work to be a general framework, simultaneously achieving age progressing (E and G), cross-age recognition (E), face morphing (G), and age estimation (Dimg).

References

[1] Z. Zhang, Y. Song and H. Qi, Age Progression/Regression by Conditional Adversarial Autoencoder, 2018. [Online]. Available: <https://arxiv.org/pdf/1702.08423.pdf>.

[2] Neural Network Learns to Synthetically Age Faces and Make them Look Younger, 2018. [Online]. Available: <https://www.technologyreview.com/s/603684/neural-networklearns-to-synthetically-age-faces-and-make-them-look-younger-too/>

[3] UTKFace, 2018. [Online]. Available: https://www.kaggle.com/jangedoo/utkface-new.

[4] G. Antipov, M. Baccouche and J. Dugelay, FACE AGING WITH CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS, 2018. [Online]. Available: https://arxiv.org/pdf/1702.01983.pdf.