Profile to Frontal Facial Generation Using Conditional GAN

Roy Park
Stanford University
Stanford, CA
rpark3@stanford.edu

Abstract

Facial recognition has arguably become one of the most common ways to unlock one's phones. The profile or the side-view of one's face is paper aims to ConvNet-based conditional Generative Adversarial Network (cGAN).

1. Introduction

Facial recognition has arguably become one of the most common ways to access various applications and functions on one's phone. For example, Apple has 'Face ID' which allows users to unlock the phone itself and also provide access to Apple Pay without having to type numerical password. In addition to providing security, facial recognition has applications for the purpose of entertainment especially amongst social media apps. Such mobile apps as Snapchat and Tiktok allows users are able to create video content using their built-in filters which are usually based on facial recognition.

Oftentimes, profile view (or the side-view) of one's face is neglected as a method for facial recognition. The previously mentioned Apple's Face ID and Snap's filters are all almost exclusively applied to frontal facial image and video data. However, the profile view of one's face is as uniquely distinguishable as the frontal view and the two views of one face are closely interconnected and related to each other. Hence, I wanted implement and experiment with generating frontal view of facial data given its profile view.

Specifically, I used conditional Generative Adversarial Network (cGAN) in order to predict the frontal facial view when given the profile view of facial data. Because larger volumes of frontal facial data and facial recognition models exist, being able to predict the frontal view when given a profile view of a facial data should allow applicability to pre-existing models for facial recognition.

2. Related Work

Sengupta recognizes the significant drop in prediction accuracy for profile (side-view) facial recognition, and they propose various methods for profile facial recognition. They various computer vision techniques that do not use

Isola proposes Conditional Adversarial Networks for general purpose image-to-image translation. Conditional Adversarial Network is a

3. Data

I plan to utilize a dataset titled "Celebrities in Frontal-Profile in the Wild". This dataset has a collection of a total of 500 celebrity faces labeled by their names and whether the image has a frontal face or a profile (side-view) face. Each celebrity had a total of 10 frontal pictures and 4 profile pictures, and I chose to pair the 4 profile pictures to 4 frontal pictures. Hence, there were a total of (4 + 4) * 500 = 4000 images consisting of 2000 profile facial images and 2000 frontal facial images.

I used Python to move my data into two different folders in which one folder contained all the . I have also changed the name of each image file to make it more convenient to pair up the input-label, which is essentially the profile-frontal pair (Fig. 1).

4. Methods

I chose to utilize conditional Generative Adversarial Network to tackle the problem of generating frontal view facial images out of profile view facial image. I could have used regular Convolutional Neural Network in which the forward feed takes in a 3x64x64 image and also generates a 3x64x64 image, but because our main purpose is to be able to generate, I chose to go with generative networks.

Because I wanted to implement each part of my code and be able to see how each part of my code works, I decided to start training them. I also chose to implement my code on a Jupyter Notebook to minimize errors and accelerate debugging process.

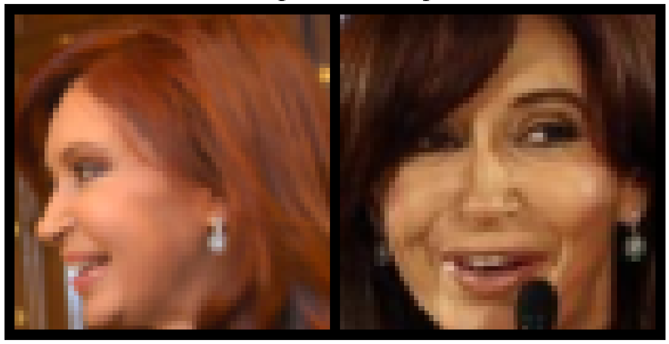

Figure 1. Example of profile-frontal pair from the dataset.

5. Experiments

I used a Convolutional Neural Network as the building block for both of my generative and adversarial network. Specifically, for the generative network, I tried various numbers of convolutional layers. I first tried 4 pairs of convolutional layer and batchnorm layer followed by 4 pairs of 2D transposed convolutional layer and batchnorm layer. For my discriminator network, I used 5 pairs of convolutional layer and batchnorm layer.

6. Conclusion

The result of my experiment is overall mixed. My model seemed to work well on certain groups of people (usually within a specific race group). For instance, we can see that my model is able to predict both Caucasian male (Fig. 3) and female (Fig. 2) fairly accurately.

On the other hand, my model did not perform too well on facial data of other races (**??**). This could be because of the my dataset is not so racially diversified.

One limitation that I encountered could be the limited size of dataset. There are a total of 4 frontal facial images and 4 profile facial images of 500 different celebrities, which amounts to a total of 4000 images. Hence, I have notices that our model overfits since it almost perfectly generates the input image from the training dataset, but fails to do so for training dataset. In the future, I could try to utilize various regularization methods such as dropout layers for my generative network to prevent my network from overfitting (Fig. 6).

References

Figure 2. Generated image of Caucasian female.

Figure 3. Generated image of caucasian male.

Figure 4. Examples of model predicting inaccurately for facial data of African-descent.

Figure 5. Examples of model overfitting training data.

Figure 6. Examples of model failing to predict accurately on test data.