CS230 Winter 2018

Image Generation and Completion

Yijun Jiang, Yi Liu, Zhengli Wang

Introduction

Generative adversarial network (GAN) consists of two networks contesting with each other in a zero-sum game framework. It has been widely applied to image generation problems.

Image completion, on the other hand, involves filling out missing parts of an incomplete image. It is an active area of research with important applications such as image editing and super-resolution. It is challenging because an algorithm not only needs to "infer" the missing parts from the neighboring pixels, but also "learn" the content from the existing ones.

In this project, we implement a GAN to generate images of human faces. We then build a completer using our trained GAN to infer missing pixels in human face images.

Fig.1 Image completion applied in restoration of old photos

Data

RGB images of human faces were obtained from the Large-scale CelebFaces Attributes (CelebA) Dataset. This dataset contains 10,177 identities, 202,599 face images, 5 landmark locations (x and y locations for left and right eyes, a nose and a mouth for each image), and 40 binary attributes annotations.

We first cropped each image from the original size of 178 x 218 pixels to a new size of 80 x 80 pixels. Every image was cropped according to the landmark locations so that the face is centered and fills the whole image. Examples are shown in Fig.2.

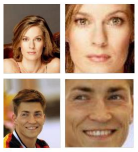

Fig.2 Original and cropped images

Then we selected faces with "good quality". Our standards for "good quality" include: 1) the image is neither too dark nor too bright; 2) not a face seen in a profile; 3) not wearing glasses. We selected 84,975 such images. Examples are shown in Fig.3.

For the image completion task, we masked a 30 x 30 pixel area at the mouth.

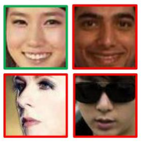

Fig.3 faces with "good quality" and "bad quality"

References

[1] Radford, Metz, and Chintala, "*Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*", ICLR 2016.
[2] Goodfellow, *et al.*, "*Generative Adversarial Networks*", arXiv:1406.2661 (2014).
[3] The GitHub of Brandon Amos, "*Image Completion with Deep Learning in TensorFlow*", http://bamos.github.io/2016/08/09/deep-completion/.
[4] DCGAN implementation: https://github.com/jacobgil/keras-dcgan

Models

To complete an image, two steps are included:

1. Generation: A DCGAN is used. The input to the generator is a length-100 vector. Both the generator and the discriminator have 4 Conv2D layers. Their structures are shown in Fig.4.

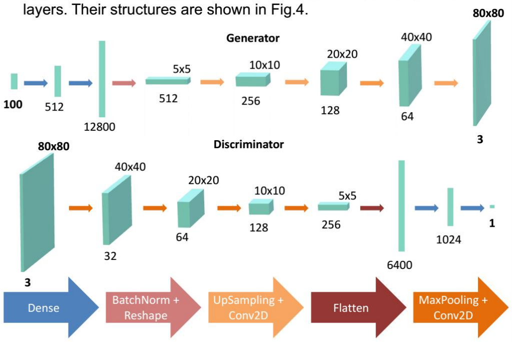

Fig.4 GAN structure

2. Completion: The image to be completed (image A) has a missing 30 x 30 pixel area. The input to the generator is updated using SGD so that the generated image (image B) resembles image A in the known part. Then generated missing pixels in image B is put back to image A. This process is shown in Fig.5.

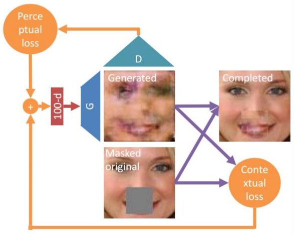

Fig.5 Completion procedure

Several notes on the models:

1. For the purpose of image generation only, we found that the generator and discriminator with more filters (up to 256) worked better at generating detailed images. However, this resulted in slow updates during the completion step. So we trained another DCGAN with less filters (up to 128 filters) for completion to speed up the process.
2. BatchNorm layers are recommended in the DCGAN paper. However, we found that too many of them hurt performance. After optimization, we only used one BatchNorm layer following the dense layer in the generator.

Results

Outputs of certain channels in each layer of the generator and the discriminator showed how the GAN works step by step in Fig.6 & 7.

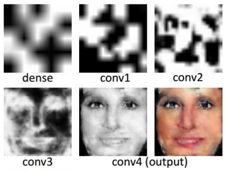

Fig.6 Outputs of selected hidden layers in the generator

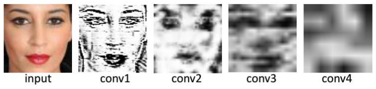

Fig.7 Outputs of selected hidden layers in the discriminator

We were able to generate human face images of a moderate quality. Details such as eyes can be reconstructed. However, the skin color is less even than real faces. Some samples are displayed in Fig.8.

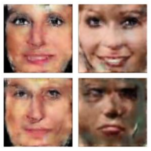

Fig.8 Generated images

Image completion results (with the smaller-sized generator and discriminator) are shown in Fig.9. Although the GAN is unable to fill in very detailed features of the face (completed pixels are usually blurry), the completions are correct perceptually.

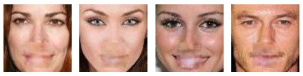

Fig.9 Completed images (pixels at the mouth are completed)

Discussion

GANs are notoriously hard to train. In this project we tested several architectures and many tricks suggested by the deep learning community. Unfortunately most trials ended up in failure. The main symptom is the training being unstable, and that the generator and the discriminator become unbalanced over time. Even with our working GAN, the generated images lack details and are often blurry. Wasserstein GAN and residue networks may provide a way to improve our performance.