Deblurring Faces with a GAN: Improving Facial Recognition and Security through Neural Networks

Aaron Lieberman
Justin Stitt
CPSC 483-01

Abstract

The project involved the use of a Generative Adversarial Network (GAN) to unblur blurred faces. The training data was generated by obtaining an external face dataset and blurring the faces to generate the pairs of blurred and unblurred faces. Tensorflow was used to create and train the neural network. This network has several potential implications in society, including improved facial recognition technology and the ability to restore faces in blurry images.

Introduction and Motivation

The motivations behind developing this project were a desire to improve facial recognition technology, a curiosity about the capabilities of GANs, and a need to restore faces in blurry images. Some potential effects on society include improved security through better facial recognition, the ability to enhance images captured on security cameras, and the potential for personal privacy concerns to arise if the technology is used without proper oversight.

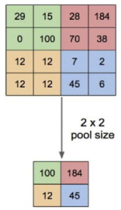
Preprocessing

We preprocess images for training a neural network by resizing them to a constant size and normalizing the pixel values to be between -1 and 1. We may also add noise and random transformations to the images to make the training data more diverse. These preprocessing steps help us prepare the images for training and improve the network's performance.

672 x 672 256 x 256

Downsampling & Upsampling

Upsampling is the process of increasing the resolution of images generated by the generator in a GAN, while downsampling is the process of decreasing the resolution of images fed into the discriminator. These processes allow the generator and discriminator to operate at different scales, providing layers of context to areas of an image, and improve the overall performance of the network.

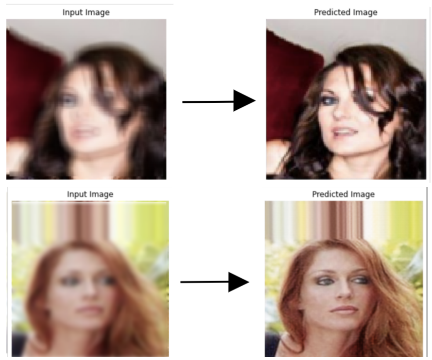


GAN: Discriminator & Generator

The discriminator and generator are two neural networks that work together to generate new data. The discriminator is trained to classify images as either real or fake, while the generator produces synthetic images that are intended to fool the discriminator. As the generator becomes better at producing realistic images, the discriminator must become better at distinguishing real from fake. This "dance" back and forth allows the GAN to generate new data that is strikingly similar to the original dataset.

Results

Given a blurred face, the model is able to generate a "predicted" image wherein it attempts to "unblur" the face.



Conclusion

One of the key positive aspects of the GAN model is that it was able to produce very realistic looking unblurred faces from blurred input images. Additionally, the use of a validation data set allowed the model to be tested on unseen images, which showed that it was able to generalize well to new data. However, it is worth noting that the model did not perform as well on images that were drastically different than the style of the face dataset used to train it. This suggests that further work may be needed to improve the model's ability to handle a wider range of image styles.

References

[0]python.plainenglish.io/how-to-read-parquet-files-in-python-without-a-distributed-cluster-4d4e8ba600e5
[1] www.tensorflow.org/tutorials/generative/pix2pix
[2] www.tensorflow.org/api_docs
[3] en.wikipedia.org/wiki/Generative_adversarial_network