Image Super Resolution for Facial Recognition

Corey Shih
cshih@stanford.edu

Abstract

The problem of super resolution entails artificially enhancing the resolution of a low resolution image to obtain a plausible corresponding higher resolution image. This is of particular use to images used for facial recognition, as higher resolution inputs can improve the performance of facial recognition systems. In this project, I train two conditional generative adversarial networks (cGANs) to output $250 \times 250$ images from $50 \times 50$ and $25 \times 25$ input images of human faces. The resulting outputs are clearer than those obtained from bicubic interpolation, but still leave room for improvement.

1. Introduction

Facial recognition systems are becoming increasingly common in day-to-day life; however, one of the biggest detractors from the performance of such technology is blurry or low resolution images. Traditional techniques for increasing image resolution, such as interpolation, usually result in blurry images. Being able to produce accurate estimates of high resolution images from low resolution ones would greatly improve the performance of facial recognition systems, and is the goal of the field of image super resolution. Image super resolution in general proves to be a challenging problem since there can be many plausible high resolution images corresponding to the same low resolution image.

This project attempts to generate $250 \times 250$ color images of human faces from $25 \times 25$ and $50 \times 50$ color image inputs of faces. The inputs are fed into a cGAN, which take either the $25 \times 25$ or $50 \times 50$ image as input, and generate a higher resolution $250 \times 250$ output image.

2. Related Work

This project was inspired by work done by Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens at Google Brain on pixel recursive super resolution.[1] In the paper, they implement an extension of PixelCNN using a new probabilistic network architecture that is trained end-to-end using a log-likelihood objective. The model is capable of generating $32 \times 32$ color face images from mere $8 \times 8$ color inputs. Instead of utilizing a CNN, I attempt to tackle a similar problem using a cGAN instead.

3. Dataset and Features

Images were taken from the Labeled Faces in the Wild (LFW) database, which provides 13233 images of 5749 different people from the web.[2] The database consists of color images that are $250 \times 250$ in resolution. I used TensorFlow to downscale these images to $25 \times 25$ and $50 \times 50$ resolution using nearest-neighbor interpolation for use in training two separate models. The low resolution and corresponding original resolution image are fed into the GAN as an input/target pair. The image pairs were split into training and test sets with 10586 and 2647 images, respectively. Figure 1 shows an example of an image from the dataset, as well as its lower resolution versions that are used as inputs to the model.
4. Methods

Two cGAN models were implemented using the pix2pix architecture by Isola et al. for 25 x 25 inputs and 50 x 50 inputs.[3]

GANs consist of two major components, the generator and the discriminator. The generator takes in the low resolution input (25 x 25 or 50 x 50) and outputs a 250 x 250 resolution image. The discriminator takes in the input/target and input/output image pairs and attempts to determine which pair is real and which is generated. The generator’s parameters are then updated based on the gradients from the loss of the discriminator. By training in this way, the generator is learning to beat the discriminator, and ideally by the end of training will be capable of generating plausible images capable of fooling the discriminator or a human.

In general, the losses of the discriminator and generator are the standard cross-entropy losses for classification problems. The discriminator loss is given by:

\[
J^{(D)} = -\frac{1}{m_{real}} \sum_{i=1}^{m_{real}} y^{(i)}_{real} \log(D(x^{(i)}))
- \frac{1}{m_{gen}} \sum_{i=1}^{m_{gen}} \left(1 - y^{(i)}_{gen}\right) \log\left(1 - D(G(z^{(i)}))\right)
\]

Here, \(D\) and \(G\) represent the discriminator and generator outputs, respectively. For this implementation, I also added a L1 loss term to the generator loss, which measures the difference between the output image and target image. In doing so, the generator is incentivized to generate images similar to the target image.

For this project, I used TensorFlow to train both cGAN models on a Tesla K80 GPU for 10 epochs, taking roughly 10 hours each. An Adam optimizer was used with a learning rate of 0.0005, \(\beta_1 = 0.9, \beta_2 = 0.999, \text{ and } \varepsilon = 10^{-8}\). A mini-batch size of 128 was used for training.

5. Results and Discussion

Some results from both the 25 x 25 and 50 x 50 models are shown in the figures below.
From Figure 2, the outputs from the $50 \times 50$ model generally look very reasonable, although they are still quite blurry when compared to the target images. The results from the $25 \times 25$ model in Figure 3 show that the outputs are much more distorted and have more artifacts, likely due to the fact that there is much less information to work with in such low resolution input images. For the most part, the cGAN seems to do a reasonable job of simulating higher resolution images, but the results are still far behind that of Dahl et al. One reason for this is due to the fact that the authors of the Google Brain paper worked with $8 \times 8$ inputs and $32 \times 32$ targets, which is a factor of 16 increase in the number of pixels. For comparison, going from $50 \times 50$ inputs to $25 \times 25$ targets is a factor of 25 increase in pixel count, and $25 \times 25$ inputs to $25 \times 25$ targets is a factor of 100 increase. The models implemented in this project need to simulate many more pixels, making the overall task more difficult.

Both models perform significantly worse on images of people with their face at an angle, likely due to the fact that there simply are not as many images of angled faces in the database. Incorporating more of such images into the training set may improve performance on such examples.

Figures 4 shows a comparison between the output images from the $50 \times 50$ model and the corresponding result from bicubic interpolation. While still a bit blurry, the outputs from the $50 \times 50$ model are noticeably clearer than the images obtained from bicubic interpolation. Edges in particular are much sharper in the model output, and jagged edges are not present. The overall level of detail in both images is still significantly less than that of the target image.
6. Conclusion and Future Work

Using images from the LFW database, I trained a cGAN based on the pix2pix architecture to generate $250 \times 250$ output images of human faces from $50 \times 50$ and $25 \times 25$ inputs. Both models resulted in clearer images than those obtained from interpolation, but were still blurry when compared to the target images. The $25 \times 25$ model exhibited significant distortion due to the lack of information in the low resolution image. Additional hyperparameter tuning and/or handicapping the discriminator may be useful in training a better model.

A number of additional methods could also be implemented to improve the performance of the models trained in this project. Additional data preprocessing such as cropping or deep funnelling could improve network performance. Exploring larger alternative datasets, such as the VoxCeleb dataset, might also yield better results. Furthermore, trying alternative network architectures or perhaps simpler CNN models could also be worthwhile.

7. Contributions

All work on this project was done by Corey Shih.

8. References

2) Huang, G.B.; Ramesh, M.; Berg, T.; Learned-Miller, E. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, *University of Massachusetts, 2007, Amherst, MA, USA.*