Super Resolution Generative Adversarial Network Magnifier

Sergio Barajas
University of Illinois at Chicago
1200 W Harrison St, Chicago, IL 60607
sbaraj5@uic.edu

Peter Hanula
University of Illinois at Chicago
1200 W Harrison St, Chicago, IL 60607
phanul@uic.edu

ABSTRACT

This paper describes the design decisions, implementation, problems, and development problems experienced in the final project of CS523 Multimedia Systems, where we implemented SRGAN to produce output images and created a User interface which takes an image and runs it through out trained network to produce and show a super resolution magnified version of the original image.

INTRODUCTION

This project consists of a trained SRGAN which can be used to produce super resolution images from lower resolution input images. Our network was trained on MNIST images for 100 Epochs, 100,000 iterations. The results are obvious and the SR image is much higher quality than the input image. We are able to control the output resolution and have attained 4K resolution. This resolution is not optimal for the MNIST data set but we believe it would be for other image datasets such as the CIFAR-100 image dataset. Our user interface allows a user to select an input image by dragging and dropping or by browsing their file explorer and selecting an image. Once an image is selected, it is run through our SRGAN code and an output super resolution is displayed in the secondary window next to the original Image. Our UI allows the user to switch between a few different magnification levels and also between standard and super resolution mode. Our implementation allows users to explore the super resolution results in great detail and is a true super resolution magnifier.

RELATED WORK

We began our thought process by reading the articles presented in class. Specifically, ***Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network***, https://arxiv.org/abs/1609.04802; and ***Generative Adversarial Nets***, https://arxiv.org/abs/1406.2661. Our implementation is based on buriburisuri's SRGAN implementation, https://github.com/buriburisuri/SRGAN. We altered and repurposed the code for our project. Our output needed to be modified as well as the means for accepting an input image to be run through the generative net. This will be discussed in greater detail later in this paper. We also made great use of the many guides in the tensorflow.org site, particularly with respect to forming input data into properly shaped tensors to be run through the GAN.

EXPERIMENTAL AND COMPUTATIONAL DETAILS

Our environment for this project consisted of Python Version 2.7 and the dependencies are as follows:
tensorflow >= rc0.11
sugartensor >= 0.0.1.7
pip install pillow
pip install matplotlib
pip install PIL
The Tensorflow SRGAN code was run on the sage machine for use of its computational power. The code still resides there inside the TeamPergio directory. The original SRGAN code implementation would pull a input image to run through the Generative network from the MNIST data site. Our implementation changed the code to take input as a png file on the local machine and reformat that image to a 28 by 28 size .png and then reshape the image to proper tensor data format. This took some time to learn and implement as the data required proper tensor rank, shape and type/dformat(32float for our purposes). Another change that was needed for out implementation was the output file created. The original implementation outputs a .png file comprised of a 10 by 12 plot of mnist images demonstrating the difference between many example images and the different levels of resolution manipulation. For our purposes we altered the code to output two images instead of one. One image demonstrates the ground truth resolution of the input image and the second demonstrates the Super resolution image. We are able to alter the size of the output images and have it outputting 4k resolution images.

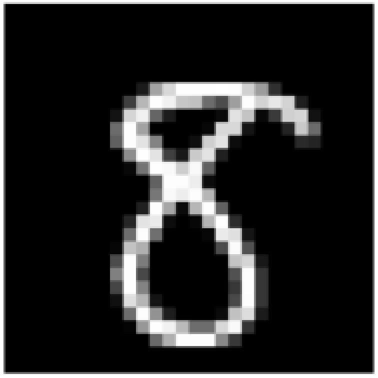

Figure 1: Ground truth output image

Figure 2: Super Resolution output image

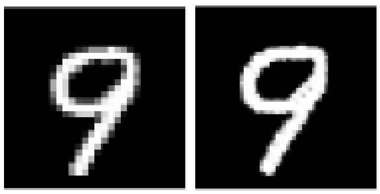

Figure 3: side by side comparison of ground truth and super resolution images

Notice the resolution differences in the images as the super resolution image is much less pixilated and noticeably higher resolution.

CONCLUSION AND FUTURE WORK

Our implementation could be trained using other image datasets. We began the process of doing this by running a python script to resize all the images in the CIFAR-100 dataset. Once this images are in proper 28 by 28 input format the dataset needs to be converted into idx/mnist file format in order to be passed to the training SRGAN. Due to time constraints and other difficulties this is not implemented at this time. It is a project designated for future work. This project tries to create a useful tool which has great potential for implementation in many different applications. One example would be using it to get a close up look for shopping applications. Another would be for inspection of hard to see details of an image, such as for detective work in criminal investigations.

REFERNCES

[1] Christian Ledig et. al. 15 Sep 2016. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial. https://arxiv.org/abs/1609.04802

[2] Ian J. Goodfellow et. al. 10 Jun 2014. Generative Adversarial Networks. https://arxiv.org/abs/1406.2661

[3] Namju Kim (buriburisuri@gmail.com) at Jamonglabs Co., Ltd. 3 Nov 2016. SRGAN(Super-Resolution Generative Adversarial Network). https://github.com/buriburisuri/SRGAN