Image Upscaling using Generative Adversarial Networks and Autoencoder

Suyash Chaudhary
Indiana University
suschaud@iu.edu

Gautam Chauhan
Indiana University
gchauha@iu.edu

***Abstract*— Image Upscaling produces high-resolution images from a set of low-resolution images. We create 2 different architectures capable of learning the inference between a high-resolution image and a low-resolution image which in turn can be used for image upscaling. GANs which have gained wide popularity after being inspired by the two-player zero-sum game can be used for upscaling images. We develop two models i.e., an autoencoder and an ESRGAN, and train them on the same dataset to perform supervised super-resolution and report that the autoencoder outperforms the ESRGAN by reporting a better PSNR by 16.66.**

Keywords—Image upscaling, generative adversarial networks (GANs), convolutional neural networks (CNNs), Super Resolution

I. Introduction

Image upscaling can produce one or a set of high-resolution images from a set of low-resolution images [1]. Deep Learning has shown superiority over traditional machine learning models in domains such as speech recognition [2], natural language processing [3], and lastly computer vision [4]. Deep learning based models can learn underlying complex patterns in unstructured data which has led to these advancements. Recently, a lot of focus has been put on minimizing the mean squared reconstruction error or increasing PSNR in the field of image upscaling. We perform supervised super-resolution or super image upscaling by developing and training two different types of novel deep learning architectures. For this, we use the famous benchmarking Berkeley segmentation dataset (BSD200) to train both of our models i.e., autoencoder model and an ESRGAN to compute various metrics and losses. This project primarily serves, as an illustration, to showcase our steep learning curve from the course.

II. Background

A. Convolutional Neural Networks

Convolutional Neural Networks or CNN are a well-known class of deep learning architectures inspired by the natural visual perception mechanism of living creatures. Researchers have been introducing depth to CNNs so that the network can better approximate the target function with increased nonlinearity and get better feature representations. CNN architectures are comprised of 3 layers i.e., a convolutional layer, a pooling layer, and a fully connected layer.

Super-Resolution Convolutional Neural Network (SRCNN) proposed by Dong et al. aims at recovering high-resolution images using any given low-resolution image [5]. SRCNNs have paved their way into solving some of the most complex research areas such as autonomous driving [9], enhancing the quality of radar images [10], and creating high-definition display technology for underwater images [11].

The proposed architecture for SRCNN by Dong et al. [5] proved to be faster than several other methods such as sparse-coding based method [6], and its improvements [7], [8]. SRCNN traditionally works by taking a low-resolution image as input and performing patch extraction and representation. Further, feature maps of a low-resolution image are created which are then mapped to a high-resolution image nonlinearly. The final high-resolution is then reconstructed. Experiments are performed by differing the architecture of the model by changing the depth, the number of filters, and filter sizes.

Zeng et al. [14] proposed a deep architecture named coupled deep autoencoder for super-resolution with the idea that the autoencoder can map the arbitrary relationships between the intrinsic relationships of low-resolution and high-resolution patches in both linear and non-linear cases. The proposed architecture works in 4 stages. In stage 1, the learning of intrinsic representation of low-resolution images is done followed by the learning of relationships of high-resolution images in stage 2. Stage 3 consists of mapping representations between low-resolution and high-resolution images. Finally, backpropagation is conducted for fine-tuning.

B. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a clever way of training a generative model by converting an unsupervised problem to a supervised problem. The model consists of two sub-models: a generative model which is trained to generate new examples and a discriminator model which critically evaluates the generated examples. As the generator model improves with training, the discriminator performance gets worse because the discriminator can't tell the difference between real and fake easily. An accuracy of 50% for the discriminator is considered to be good.

With large-upscaling factors, it was complicated to use CNNs for image upscaling problems. This problem was solved by researchers at Twitter by proposing a generative adversarial network for super-resolution (SRGAN) [13]. The proposed network had a high perceptual quality and improves the reconstructions by moving toward the regions of the search space. The architecture of generator and discriminator in SRGAN

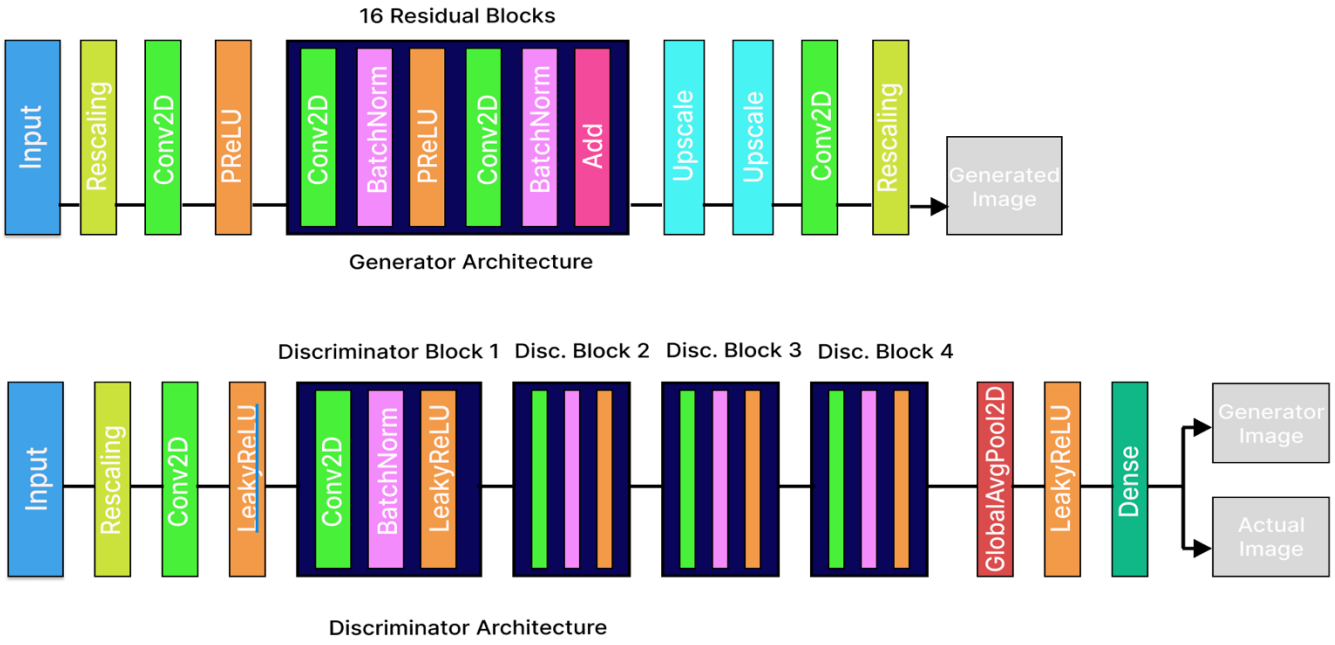

Figure 1: ESRGAN

comprised of residual blocks with skip connections for an optimized perceptual loss. Two convolutional layers with small 3 X 3 kernels along with 64 feature maps were used. Furthermore, we added batch normalization with a fixed momentum value was used and ParametricReLU was added as activation function.

The Super-Resolution Generative Adversarial Network (SRGAN) proposed by Wang et al. [12] introduced Residual-in-Residual Dense Block (RRDB) without batch normalization as the basic network-building unit. These RRDBs were easier to train and enhanced the complexity of the model which in turn boosted the performance. RRDBs are where the residual learning takes place at different levels in the architecture. The discriminator introduced in the architecture of ESRGAN enhanced the structure based on Relativistic GAN. A Relativistic GAN tries to predict the probability that a real image is better than a fake image. But as seen in Figure 1, the residual block still has batch normalization layer because unlike the SRGAN architecture proposed by Wang et al. [12] which uses Leaky ReLU for activation, we are using Parametric ReLU or PReLU.

III. Data & Methodology

We use the publicly available Berkeley Segmentation Dataset (BSD200) which consists of 100 low-resolution and 100 high-resolution images [15]. BSD200 has been widely used as a benchmark dataset by researchers to showcase their work and show the results.

With the idea of creating two architectures and comparing the results on the BSD200 benchmark dataset, we create two models i.e., a deep autoencoder and an SRGAN.

A. Deep Autoencoder

With the promises shown by autoencoders in unsupervised learning, a deep autoencoder model with the ability to cumulatively learn the relationships in low-resolution and high-resolution images seemed feasible.

Algorithm: Training for Deep Autoencoder
Input: Patch sets from LR/HR images $\mathbf{X} = \{x_1, \ldots, x_n\}$ and $\mathbf{Y} = \{y_1, \ldots, y_n\}$ **Output:** HR Image **Step 1**: Initialize W_1 weight and B_1 bias by learning intrinsic representations of low-resolution image patches Y. **Step 2**: Initialize W_2 weight and B_2 bias by learning intrinsic representations of high-resolution image patches Y. **Step 3**: Perform non-linear mapping of intrinsic representations of low-resolution images and high-resolution images. **Step 4**: Tune the model using backpropagation to get high-resolution image patches. **Step 5**: Reconstruct the high-resolution image.

Our deep autoencoder model contains two convolutional blocks of size 32 each having 32 filters of size 6x6, initialized by HeUniform and regularized by a small constant value of 10e-10, which is followed by two more convolutional blocks of size 64 having 32 filters of size 3x3 along with which two more convolutional blocks of size 128 and same parameters as the previous blocks were added which is followed by a 256 sized convolutional block. For the deconvolution process, a

Conv2DTranspose block was added and three more blocks of size 128, 64, and 64 respectively were included to get the output image.

B. ESRGAN

The ESRGAN model that we created is an extension of the proposed architecture by Wang et al. [12]. The purpose of a generative model G in our model is to fool the differentiable discriminator D. 16 residual-in-residual blocks with identical layouts composed of two convolutional layers with small 3 X 3 kernels, batch normalization layers, parametric rectified linear units as the activation functions are present. As more layers and more connections can boost the performance of a model, the residual-in-residual structures of our model benefit the network capacity because of the dense connections.

We use the perceptual loss as proposed by Wang et al. [12] to report the accuracy of our ESRGAN which provides stronger supervision for brightness consistency and texture recovery. The perceptual loss was initially proposed to enhance the visual quality of the image by minimizing the error in a feature space instead of the pixel space. Furthermore, our training step uses other metrics like PSNR and pixel loss. Lastly, as part of a custom training process, we modify our gradient using gradient tape and we feed the aggregate of Generator loss, perceptual loss, and pixel loss as gradient_total_loss to our tape.

IV. Results

As image upscaling techniques require losses with which we can differentiate the high-resolution image from the low-resolution image, losses such as pixel loss based on MSE, or Peak signal-to-noise ratio (PSNR) can be used.

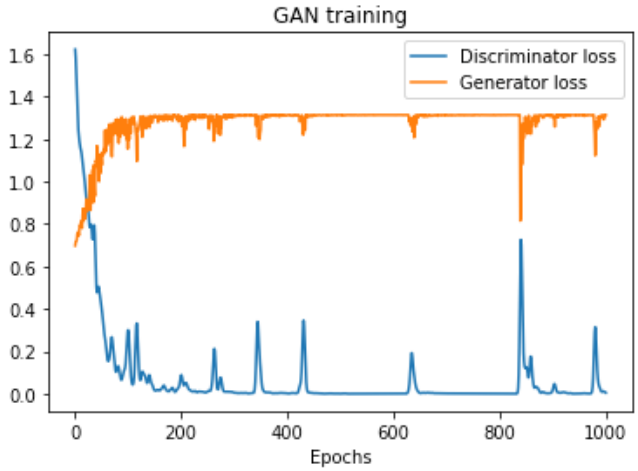

Figure 2: GAN Training

We store the generator loss, discriminator loss, perceptual loss, pixel loss, and the peak signal-to-noise ratio (PSNR) for our ESRGAN and report them. The peak signal-to-noise ratio is generally used for quality estimation and is based on MSE. Figure 2 shows the losses for both the generator and discriminator. The ESRGAN also achieves an average PSNR value of 23.81. Figure 3 shows the PSNR plotted on the y-axis with the x-axis depicting the number of epochs we ran the model for i.e., 1000.

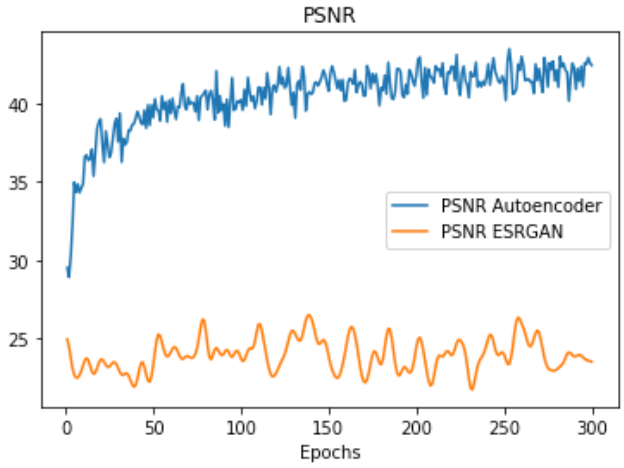

Figure 3: PSNR

Autoencoder architecture achieves a mean PSNR value of 40.47. Comparing the mean PSNR values of Autoencoder and ESRGAN, the autoencoder and ESRGAN have a difference of 16.66. The perceptual loss of ESRGAN is 0.13 and the final loss reported by the autoencoder is 0.05.

References

[1] Dong, C., Loy, C.C., Tang, X. (2016). Accelerating the Super-R…G. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] R.Collobert and J.Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proc. ACM 25th Int. Conf. Mach. Learn., 2008, pp. 160–167.

[3] A.Krizhevsky, I. Sutskever, and G. goog E. Hinton, "Image net classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Pro- cess. Syst., 2012, pp. 1097–1105.

[4] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang, "Image Super-Resolution Using Deep Convolutional Networks", Computer Vision and Pattern Recognition (cs.CV); Neural and Evolutionary Computing (cs.NE), 2015.

[5] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, Thomas Huang, "Deep Networks for Image Super-Resolution with Sparse Prior", Computer Vision and Pattern Recognition, 2015.

[6] Yang, J., Wang, Z., Lin, Z., Cohen, S., Huang, T.: Coupled dictionary training for image super-resolution. IEEE Transactions on Image Processing 21(8), 3467–3478 (2012)

[7] Timofte, R., De Smet, V., Van Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution. In: IEEE Asian Conference on Computer Vision (2014)

[8] J. Kim *et al.*, "Performance Comparison of SRCNN, VDSR, and SRDenseNet Deep Learning Models in Embedded Autonomous Driving Platforms," *2021 International Conference on Information Networking (ICOIN)*, 2021, pp. 56-58, DOI: 10.1109/ICOIN50884.2021.9333896.

[9] Y. Dai, T. Jin, Y. Song, H. Du and D. Zhao, "SRCNN-Based Enhanced Imaging for Low Frequency Radar," *2018 Progress in Electromagnetics Research Symposium (PIERS-Toyama)*, 2018, pp. 366-370, DOI: 10.23919/PIERS.2018.8597817.

[10] Y. Li *et al.*, "Underwater Image High Definition Display Using the Multilayer Perceptron and Color Feature-Based SRCNN," in *IEEE Access*, vol. 7, pp. 83721-83728, 2019, DOI: 10.1109/ACCESS.2019.2925209.

[11] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, Xiaoou Tang, “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks”, Computer Vision and Pattern Recognition, 2018.

[12] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”, Computer Vision and Pattern Recognition, 2017.

[13] K. Zeng, J. Yu, R. Wang, C. Li and D. Tao, "Coupled Deep Autoencoder for Single Image Super-Resolution," in *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 27-37, Jan. 2017, DOI: 10.1109/TCYB.2015.2501373.

[14] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/