Super-resoltion with SRGAN

Ma Liang ; Xiong Zhihao
Student, Group 9
Technical University of Munich
Professorship of Multiscale Modeling of Fluid Materials
Boltzmannstraße 15, 85747 Garching

Lab course "Hands-on Deep Learning"

Abstract: In this project, we present SRGAN, a generative adversarial network (GAN[7]) for image super-resolution (SR). To achieve this, we propose a perceptual loss[5] function which consists of an adversarial loss and a content loss. The adversarial loss pushes our solution to the natural image manifold using a discriminator network that is trained to differentiate between the super-resolved images and original photo-realistic images. In addition, we use a content loss motivated by perceptual similarity,which extracted by pretrained VGG19,instead of similarity in pixel space.At the same time, we also use different upsampling methods (Transposed Conv and Pixel Shuffler) in the decoding stage to verify more efficient network structures. The final generated image can achieve good results.

I. Introduction

SRGAN is the first framework capable of inferring photo-realistic natural images for 4x upscaling factors.We used perceptual loss according to [1] and adopted feature loss instead of pixel-level loss. The entire GAN network is constructed through Generator(deep residual network[2] + Upsampling) and Discriminator(VGG network). Use methods such as Batch-normalization[4], PReLU, etc. to improve the generalization and capacity of each network.

II. Methods

A. Deep Learning Architecture

Our overall framework is based on the generative adversarial network(GAN).We employ a generator containing encoder and decoder structures to generate HR images form LR images and employ an adversarial network to distinguish the generated images from the real images.

A.1. Generator

In the generator we first obtain the latent space using an encoder with a residual network[2][3],see fig.1. The residual network is mainly composed of residual blocks, which are composed of convolutional layer with kernel-size 3 and stride 1, batch normalization layer and nonlinear layer(PReLU). The image resolution is maintained in the

forward process,only increase the channels and get latent features of the pictures,The decoder is then used to obtain the resulting high-resolution image by Upsampling methods. In the network we use some deep learning techniques to improve the network capacity.First,it was shown that deeper network architectures can be difficult to train but have the potential to substantially increase the network's accuracy as they allow modeling mappings of very high complexity. To efficiently train these deeper network architectures, batch-normalization[4] is often used to counteract the internal co-variate shift. Deeper network architectures have also been shown to increase performance for single image super-resolution(SISR),formulate a recursive CNN and present state-of-theart results. Another powerful design choice that eases the training of deep CNNs is the recently introduced concept of residual blocks and skip-connections. Skip-connections relieve the network architecture of modeling the identity mapping that is trivial in nature, however, potentially non-trivial to represent with convolutional kernels.The input to the Generator is a low-resolution image.

A.2. Discriminator

In the adversarial network we use a network structure similar to VGG,see fig.1. A block is formed by a combination of convolutional layers with 3x3 kernel-size, batch-normalization layers, and nonlinear layers named PReLU. The blocks are stacked to extract the input features, every two blocks we will use a convolution with stride of 2 to reduce the resolution of the image and increase the number of channels in the image.And the resulting backbone is input to the two-layer(1024-1) fully connected layer through dimension change and through a sigmoid layer finally gets a probability score, where 1 represents the real image and 0 represents the generated fake image. The input of the adversarial network is the real image and the fake image generated by the Generator.

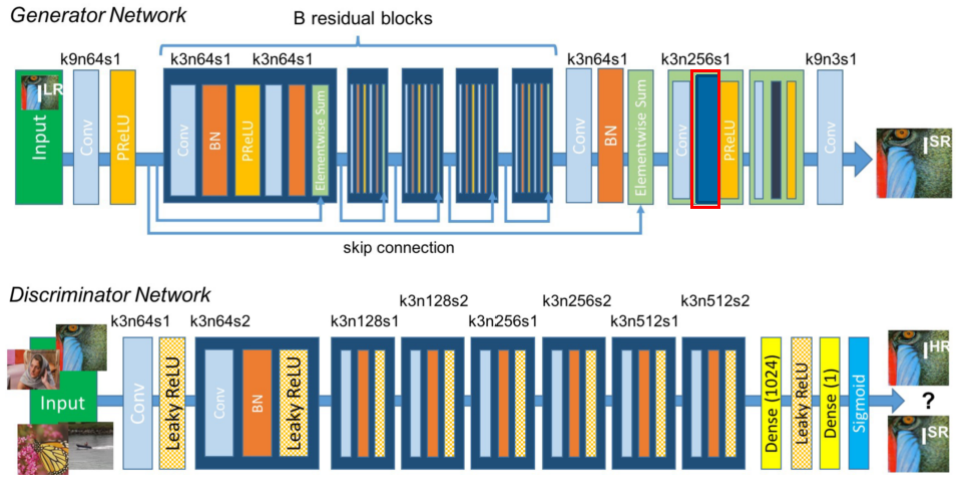

Figure 1: Architecture of Generator and Discriminator

A.3. Loss function

In this work we will specifically design a perceptual loss l^{SR} as a weighted combination of several loss components that model distinct desirable characteristics of the recovered.The definition of our perceptual loss function l^{SR} is critical for the performance of our generator network. While l^{SR} is commonly modeled based on the MSE , we improve and design a loss function that assesses a solution with respect to perceptually relevant characteristics. We formulate the perceptual loss as the weighted sum of a content loss

l_x^{SR} and an adversarial loss component as:

$$l^{SR} = \underbrace{\underbrace{l_X^{SR}}_{content\ loss} + \underbrace{10^{-3} l_{Gen}^{SR}}_{adversarial\ loss}}_{perceptual\ loss (for\ VGG\ based\ content\ losses)} \tag{1}$$

Content loss The pixel-wise MSE loss is calculated as:

$$l_{MSE}^{SR} = \frac{1}{r^2 W H} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2 \tag{2}$$

This is the most widely used optimization target for image SR on which many state-of-the-art approaches. However, while achieving particularly high PSNR,solutions of MSE optimization problems often lack highfrequency content which results in perceptually unsatisfying solutions with overly smooth textures.Instead of relying on pixel-wise losses we use a loss function that is closer to perceptual similarity. We define the VGG loss based on the ReLU activation layers of the pre-trained 19 layer VGG network. With $\phi_{i,j}$ we indicate the feature map obtained by the j-th convolution(after activation) before the i-th maxpooling layer within the VGG19 network, which we consider given. We then define the VGG loss as the euclidean distance between the feature representations of a reconstructed image $G_{\theta_G}(I^{LR})$ and the reference image I^{HR}:

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \tag{3}$$

Here $W_{i,j}$ and $H_{i,j}$ describe the dimensions of the respective feature maps within the VGG network.

Adversarial loss In addition to the content losses described so far,we also add the generative component of our GAN to the perceptual loss.This encourages our network to favor solutions that reside on the manifold of natural images,by trying to fool the discriminator network.The generative loss l_{Gen}^{SR} is defined based on the probabilities of the discriminator $D_{\theta D}(G_{\theta G}(I^{LR}))$ over all training samples as:

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{4}$$

Here,$D_{\theta D}(G_{\theta G}(I^{LR}))$ is the probability that the reconstructed image $(G_{\theta_G}(I^{LR}))$ is a natural HR image.For better gradient behavior we minimize $-\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$ instead of $\log\left[1 - D_{\theta_D}(G_{\theta_G}(I^{LR}))\right]$

B. Dataset

The dataset we used is Celeba. The wohle dataset is too large with totally more than 20,000 images. Because of the compute efficiency, it takes too much time to download and extract the whole dataset. So we decide to shrink the dataset.And for training process contains 2000 images, for validation and testing process each 250 images. For all images, we need to do preprocessing before training. The original image size is 178*281, while for the input of neural network, the images are low resultion with size 44*55. Finally the labels are high resolution images with size 176*216. Followings are the images of original image, low resolution image and high resolution image.

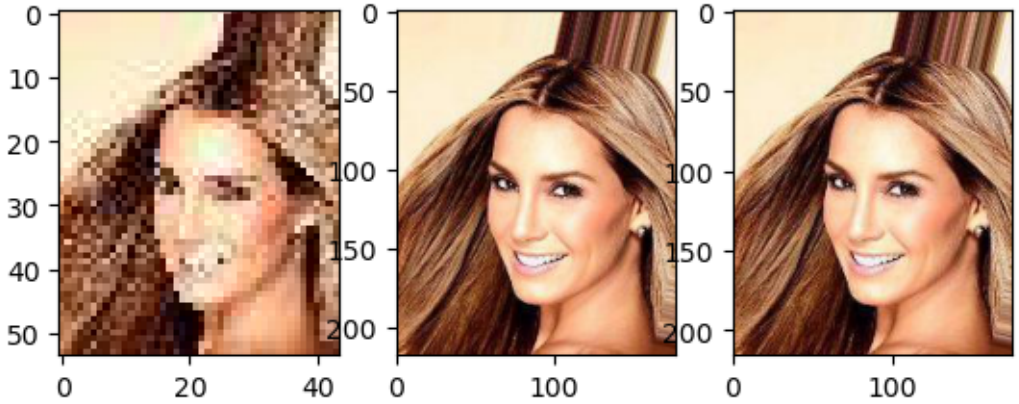

Figure 2: Left:Low resolution, middle:High resolution, Right:Original

C. Training Procedure

By training procedure, we compared two different models for SRGAN. The first training procedure is for the default SRGAN model and it was trained only by 15 epochs. For the default SRGAN, the wohle procedure was split into three training process with different upsampling structure. For the initialisation the weights of generator and discriminator are normalized with different mean value and variance. The optimization method is Adam.

The loss function is a new function called Perceptual loss function. The first five epochs are trained with structure interpolation and transposed convolution neural network. Next five epochs are trained with structure two transposed convolution neural network, and with the pretrained model from the 5th epoch. The last five epochs are trained with VGG method initialized by the pretrained model from 10th epoch. Compared with the training procedure of default SRGAN, the training procedure of SRGAN with pixel shuffle is similar. The difference is that with pixel shuffle the generator and discriminator trained directly from 0 to 15 epochs.

III. Results

A. Training Convergence

For the default SRGAN model, the training results are not good. The loss increases after 5 epochs changing the structure, and after 10 epochs, the loss even increase hundreds times bigger than first 10 epochs, which makes the test results worse. The generated super resolution images are blurred, so we can not tell any performance from these images.

But for the SGRAN model with pixel shuffle, the loss converges well. So the generated super resolution images are much better and we can see the good performance within 15 training epochs.

B. Performance

For default SRGAN, the performance is not good. But for SRGAN with pixel shuffle, the performance within 15 epochs are better.After training and validation, the test result is showed Figure 3 and 4.

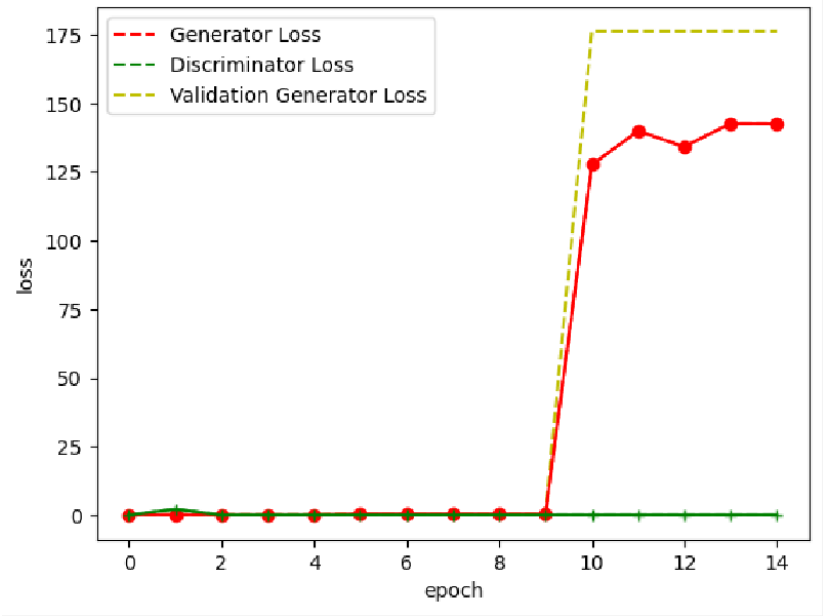

Figure 3: loss of default SRGAN

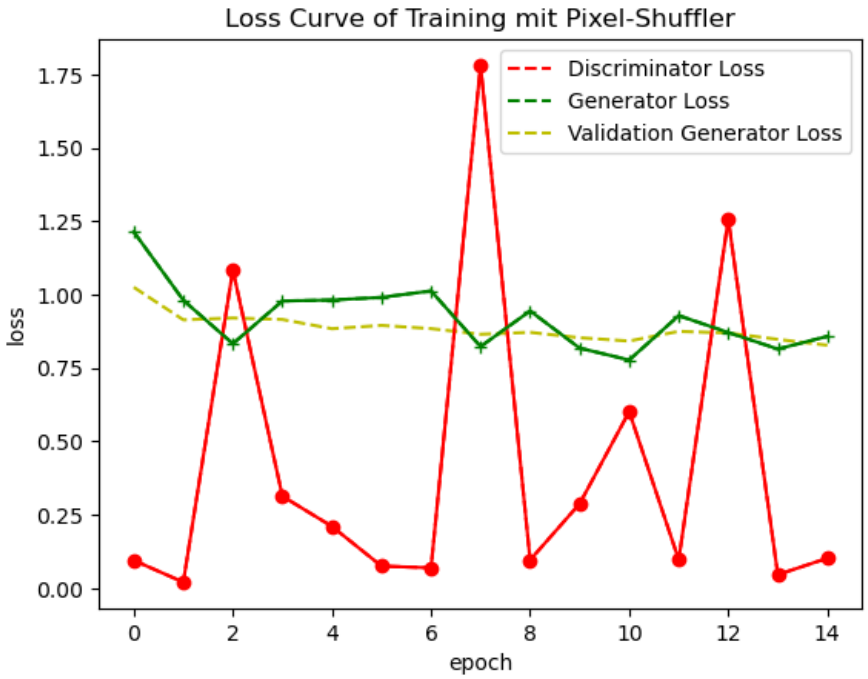

Figure 4: loss of SRGAN with pixel shuffle

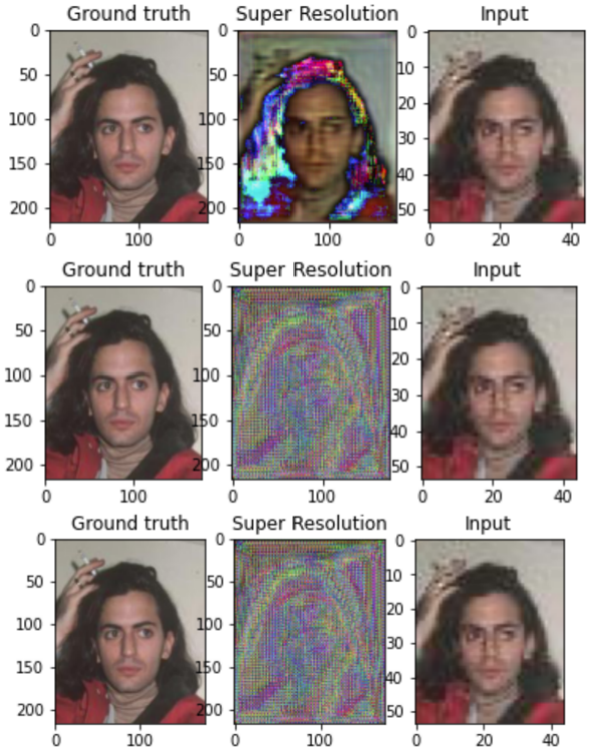

Figure 5: Performance of defalut SRGAN

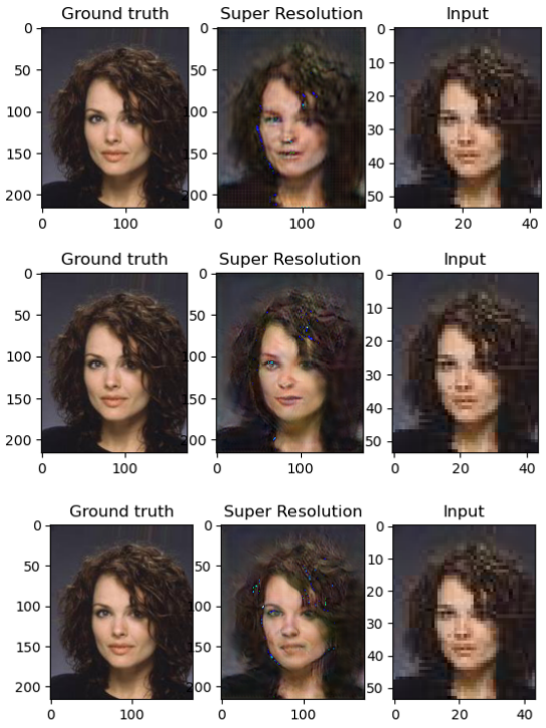

Figure 6: Performance of SRGAN with pixel shuffle

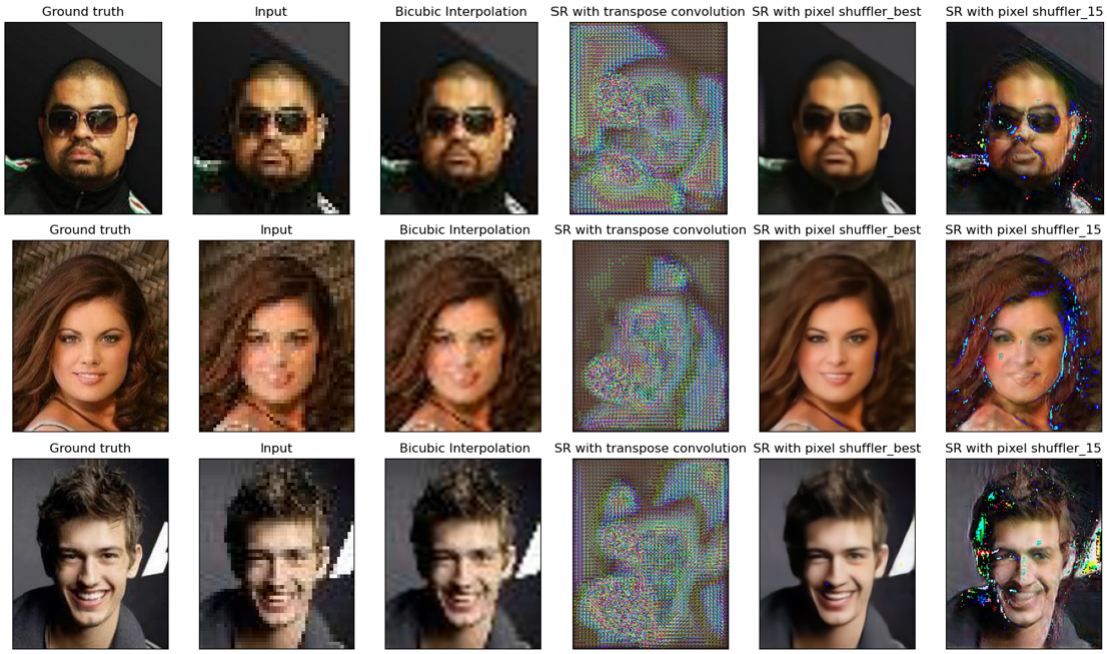

Figure 7: Test Results

C. Problems and Reasons

During the training process with default SRGAN, the images are blurred after changing the structure of Neural Network after 5 epochs. For the first 5 epochs we use interpolation and transposed convolution neural network for upsampling the super resolution images. From 5 epoch to 10 epoch, these are changed to two transposed convolution neural network. From 10 to 15 epochs, VGG is used to processing the characters of images. While the training epochs are not sufficient and the training datasets are limited by totally 2000 images. After changing the structure, the neural network can not be trained to get the well performed parameters within 5 epochs. What's more, if we just train the model using interpolation and transposed convolution neural network for 15 epochs, the results becomes better and we can see the tendency to optimize the results. And we train the model using two transposed convolution layers or using VGG for 15 training epochs. The images are still blurred. Therefore, it is confused that why the default model changed after each 5 training epochs, if the problem is not related to the training datasets and training epochs.

IV. Discussion

For the results we can see that the SRGAN with pixel shuffle model get relativ good results within 15 epochs compared with other models, such as bicubic interpolation model and the default SRGAN models. Pixel Shuffle is an operation used in super-resolution models to implement efficient sub-pixel convolutions with a stride of $\frac{1}{r}$ where r is an upscale factor. Specifically it rearranges elements in a tensor of shape (*,C x r^2, H, W) to a tensor of shape (*, C, H x r, W x r). And if we continue training the SRGAN with pixel shuffle model, we can get the best SRGAN model, which obtains much better performance and the super solution images are close to the ground truth.

V. Conclusion

To conclude, the SRGAN with pixel shuffle is a good model for the super resolution problem. But for these training and testing, much more training datasets and more training epochs are needed to obtain a good performance. Further more, this experiment gives us a new way to training the neural network. One way is that, we can change the structure of model and using the pretrained model to get a better results. Another way is that, the loss function can be chosen not only related to the traditional loss function but also some optimization with other parts related to the structure of model, such as the adversarial loss in this situation.

VI. Contributions

Student Ma Liang: Neural Network training and Network structure adjustment, PPT and Final Report

Student Xiong Zhihao: Dataset Processing, Neural Network training, PPT and Final Report

References

[1] C. Ledig et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 105-114, doi: 10.1109/CVPR.2017.19.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016

[3] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In European Conference on Computer Vision (ECCV), pages 630–645. Springer, 2016

[4] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of The 32nd International Conference on Machine Learning (ICML), pages 448–456, 2015

[5] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super- resolution. In European Conference on Computer Vision (ECCV), pages 694–711. Springer, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), pages 1097–1105, 2012.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems (NIPS), pages 2672–2680, 2014.