Group number: DL05

Name: Bharat Kashyap Karri ID: 2020AAPS0319H

CSF425 Deep Learning

Super Resolution using GAN

Abstract

Generative Adversarial Networks (GANs) in supervised datasets can produce photo-realistic images from low quality images [2]. The main challenge in this application of super resolution is for the model to create details that are lost in the lower quality images. These are referred to as high frequency details. Sharp variations of high resolution images are lost in the low resolution images. The model learns the shape, colour and texture of an image. This idea is extended by providing more information for the backpropagation of the network. It is shown how having labels in the dataset can build a more robust model with the same generator and discriminator models. A similar architecture was used in Auxiliary Classifier GANs [3] which was conditional image synthesis from a latent vector.

Introduction

The problem of super resolution is an age-old challenge that has been confronted using traditional statistical and mathematical methods, dynamic imaging and in more in the past decade, machine learning techniques. In 2017 GANs were first used in this field. GANs are a subcategory of deep neural networks introduced originally by [1] where 2 networks work in a min max operation.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

In the field of super resolution, the input image is a low resolution image and the generated image is a high resolution image. Applications of super resolution are in various fields such as biomedical imaging, using properties of materials to create images beyond the limit of resolution; surveillance, forensic, satellite imaging etc. It can also be used for better communication. Super resolution in communication can help save bandwidth for more efficient transmission.

Related Work

Most GAN architectures use a latent vector to generate images (unsupervised). In the original GAN architecture, 2 networks were used: one for generating the images called the generator and one to distinguish between real and fake images called the discriminator. The generator continuously tries to reduce a function while the discriminator tries to increase that function. The original cost function used was

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

which was modified to prevent saturation. In the problem of super resolution the data is not unsupervised anymore. The dataset consists of low resolution images and high resolution images. Low resolution images can be obtained from high resolution images by different mathematical operations. Although it may not be realistic low quality images, the results are reliable. Image to image translation models are another example of supervised learning. CycleGANs is a model used for image to image translation where it was trained without a supervised dataset, although it is a form of image to image translation.

Super resolution GAN

The objective of the model is to approximate a generator function (G(image)) such that the input low resolution image (I_LR) is mapped to its output high resolution image (I_HR). The discriminator (D(image1, image2)) aims to differentiate between real and fake images along. The net loss that the model is being trained on is the sum 2 types of losses as described in the paper during the training of the networks: the content loss and the adversarial loss. Content loss can be modelled as an MSE_loss or a VGG loss. VGG loss is a better measure than a direct MSE_loss for training as the feature maps obtained by deep layers of the VGG network are closer in representing perceptual similarity than pure MSE loss. VGG loss is the MSE loss of the feature maps obtained by passing both G(I_LR) and I_HR through the VGG network and get the desired feature map. The architecture of the generator consisted of residual blocks each made of a convolution layer of kernel size 3 and padding 1 followed by batch normalisation parametric Relu activation, a residual connection is present after another convolution and batch normalisation blocks. Number of kernels for each convolution layer is 64. 16 such residual blocks were used. The architecture for the discriminator is made up of convolution layers with kernel size 3 and the number of kernels increases by a factor of 2 from 64 to 512 as in the VGG network. Stride values are changed whenever the number of features are increased. The final 512 feature maps are followed by 2 dense layers and a final sigmoid activation to obtain a probability for sample classification.

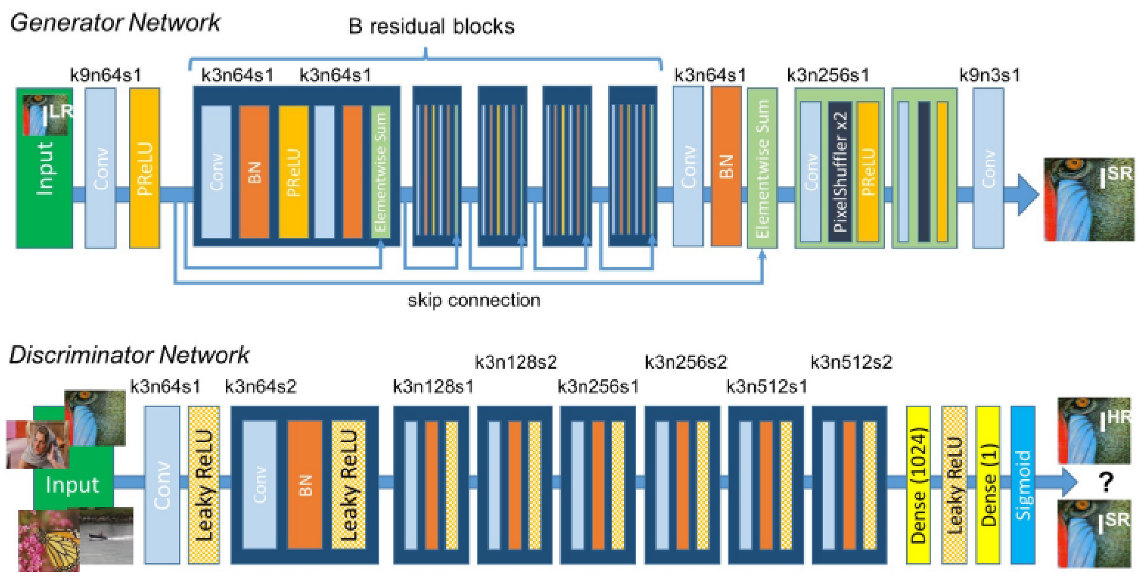


Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

The proposed model was used to upscale images by 2 times and by 4 times with 4 times upscaling the main focus.

Present Investigation

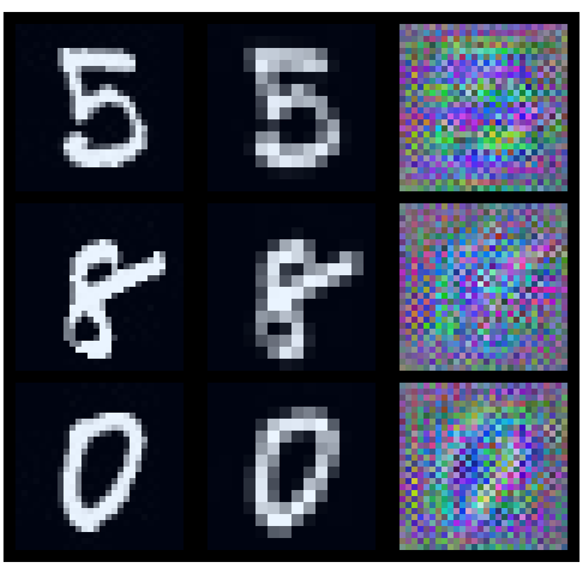
The dataset selected for the implementation of the SRGAN and proposed improvements is the MNIST handwritten digits dataset. This is a labelled dataset with each image of size 28 by 28. Each image also has a corresponding label as to which number is represented in the image. In this version of the SRGAN few changes were made to the architecture: only 4 residual blocks were used due to computational limitations; and the final 2 dense layers along with the sigmoid activation were not used. The low resolution image was generated using a BiCubic interpolation. The original idea was to compare the classification of the high resolution image and the generated image using a pretrained convolutional network. The cross entropy loss from the CNN was then added to the discriminator and generator losses. The additional loss term did not change the results by a lot and thus returned similar results. The lack of change is speculated to be due to the lack of new information. Any information derived from the CNN is similar to the information derived from the VGG feature maps. However it can be seen that the results from the modified SRGAN reached a clearer image faster than the original.

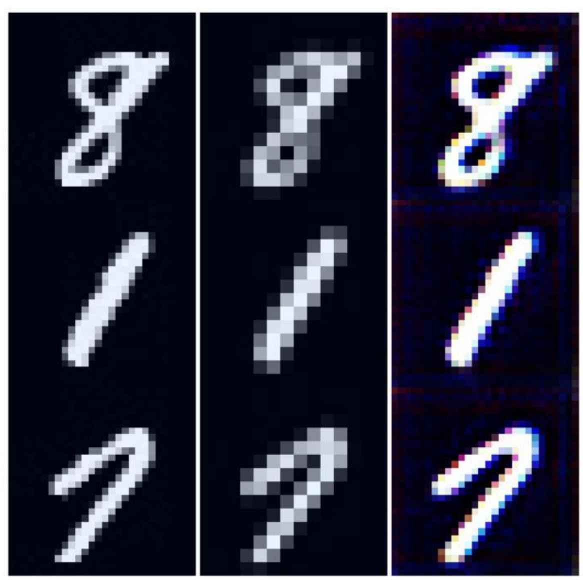
Experiments and results

Implementations of SRGAN and SRGAN using an additional pre trained CNN

SRGAN

Dataset contained 20000 images for training. A batch size of 128 was used along with the details mentioned above. The followed results were obtained. The leftmost result is the ground truth high resolution image. The middle image is the low resolution image and the rightmost image is the generated image. Since 3 channels were used for a black and white image, there are areas of different colours around the number.

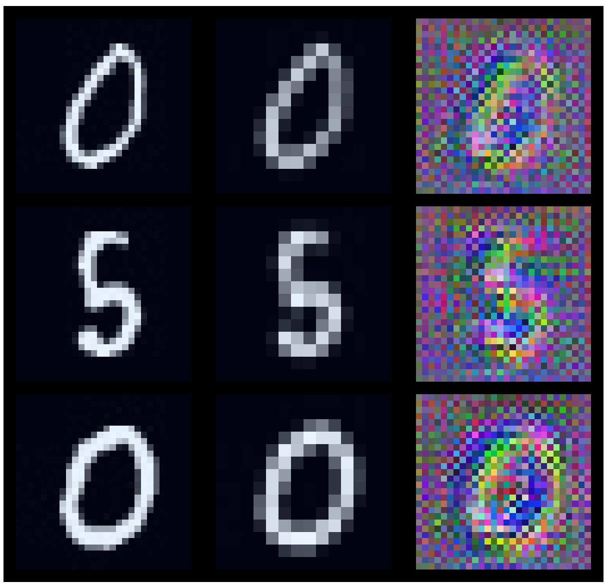


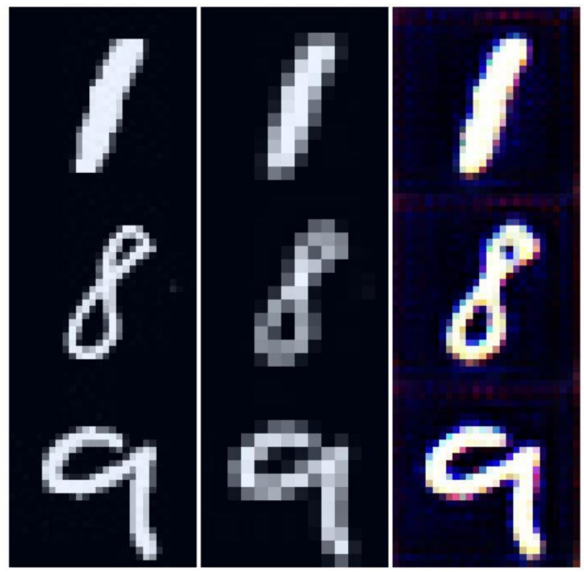


The left image shows the results after 10 iterations whereas the right image shows the results after 150 iterations.

SRGAN with pretrained CNN

Here the CNN loss was added to the generator and discriminator while training. The weight with which the CNN loss was added was considered to be a hyperparameter.





Conclusion

SRGAN has produced very realistic results on the MNIST dataset. The additional CNN loss allowed for the model to reach clearer images faster than the original. However applications in super resolution often do not have labelled images and so this improvement is very limited to such datasets.

However adaptations of this type of network could be done to higher resolution and more complex images using embeddings from text that describe the image as the class of the image. This way there would be a more robust content comparison present for the network and thus faster and more robust models.

References

1. Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.
2. Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
3. Odena, Augustus, Christopher Olah, and Jonathon Shlens. "Conditional image synthesis with auxiliary classifier gans." *International conference on machine learning*. PMLR, 2017.

Github link

The link contains implementations of SRGAN, modified SRGAN and the MNIST CNN networks. Results are also present for some of the tested values of hyperparameter of the modified SRGAN.

https://github.com/Bharat1252003/CS-F425-Deep-Learning