Super Resolution using Deep Neural Network and GANs

Hao Wang, Yinrui Li, Taiyu Dong

Introduction

Super-resolution is a set of techniques to upscale images or videos. Based on the input data, we can divide existing methods into two categories: generate high-resolution image from single low-resolution image; recover high-resolution image from multiple images. For our project, we focus on single image super resolution. The central problem of single image super resolution is how do we generate decent texture detail when we super-resolve at large upscaling factors.

The most common way to solve the problem mentioned above is maximizing the peak signal-to-noise ratio and minimizing mean squared error between generated high-resolution image and the ground truth.[1] However, signal-to-noise ratio and mean squared error are hard to capture perceptually differences. Thus, we define a proper perceptual loss function to help distinguishing the differences between generated high-resolution image and ground truth. In our project, we employ a naive deep residual network generator with mse error to the ground truth, then elaborated to a more sophisticated model based on the idea of generative adversarial network.

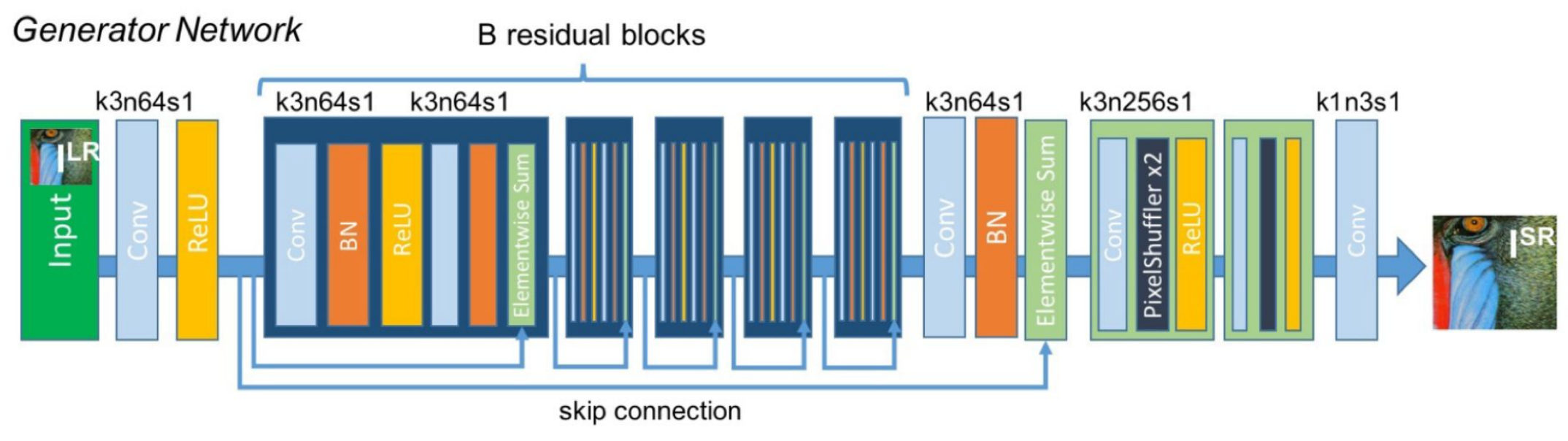
Data & Generator

Data Set:

For the training and test dataset, we use DIV2K bicubic dataset from NTIRE 2017.

Generative Model:

In both of the two method we used a same generated network architecture that takes low resolution images as input



The CNN design architecture is modified from Ledig et al., (2016) [1], where k3n64s1 means there are 64 number of 2D kernels with size 3x3 and stride 1. There are total 16 B residual blocks with skip connection by element-wise sum. The PixelShuffler x2 is a sub-pixel convolutional layer proposed by Shi et al.,(2016)[2] to increase the image resolution by a factor of 2 in both width and height.

Deep Convolutional Neural Network

Loss function:

In this method, we use the generative network above and define the loss function(mean squared error) as:

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

Training Detail (parameter):

Batch size: 32
Adam Optimizer with BETA1: 0.9
Learning rate: 1x10^{-4}
Input image size: 56 x 56 x 3
Output image size: 224 x 224 x 3
Epochs of training: 300

Generative Adversarial Network

To improve our generative model, we design new proper loss function and discriminator network. Based on the idea of generative adversarial network[3], we define a discriminator network *D* to distinguish generated image and ground truth, and a generator network *G* to mislead discriminator network *D*. Thus, we have the adversarial min-max equation for *HR/LR* represents high/low resolution image and *I* represents input data:

$$\min_{\theta_G}\max_{\theta_D} \mathbb{E}_{I^{HR}\sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR}\sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))]$$

Loss function:

We combine content loss, which includes mean squared error and VGG(19), and adversarial loss as general perceptual loss function, which is the loss function used to train the generative network.

(1)Content loss:

①MSE: same as deep convolutional neural network
②VGG(19):

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

(2)Adversarial loss:

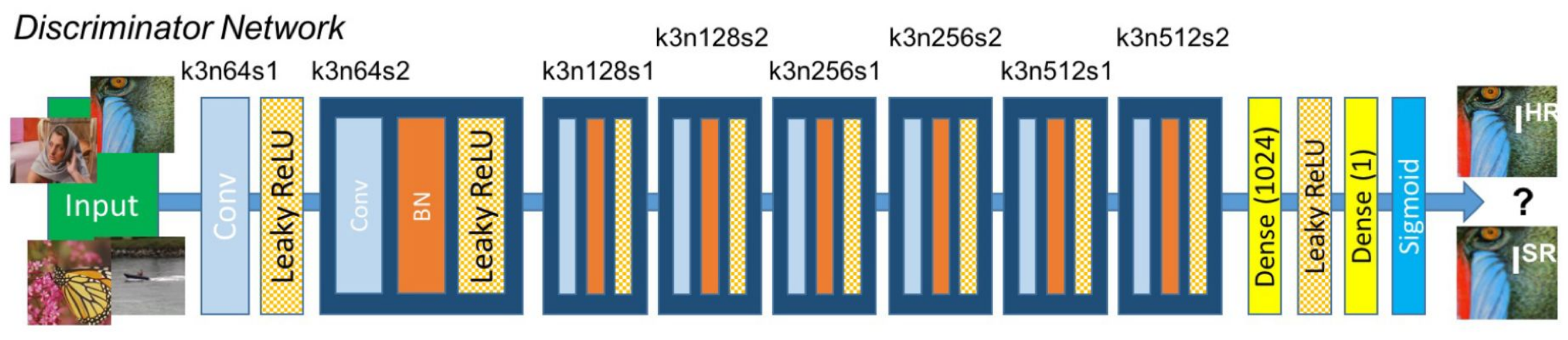
$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

We combine content loss and weighted adversarial loss as our general perceptual loss function:

$$l^{SR} = l_{MSE}^{SR} + l_{VGG}^{SR} + 10^{-3}l_{Gen}^{SR}$$

Discriminative mode:

The architecture of discriminative model is shown below. It’s modified from Ledig et al., (2016) [1]



Loss function of discriminator:

Discriminator aims to distinguish ground truth and generated images. Thus, we define loss function as follow:

$$l_G^{SR} = -\sum_{n=1}^{N} logD_{\theta_D}(1 - G_{\theta_G}(I^{LR}))$$

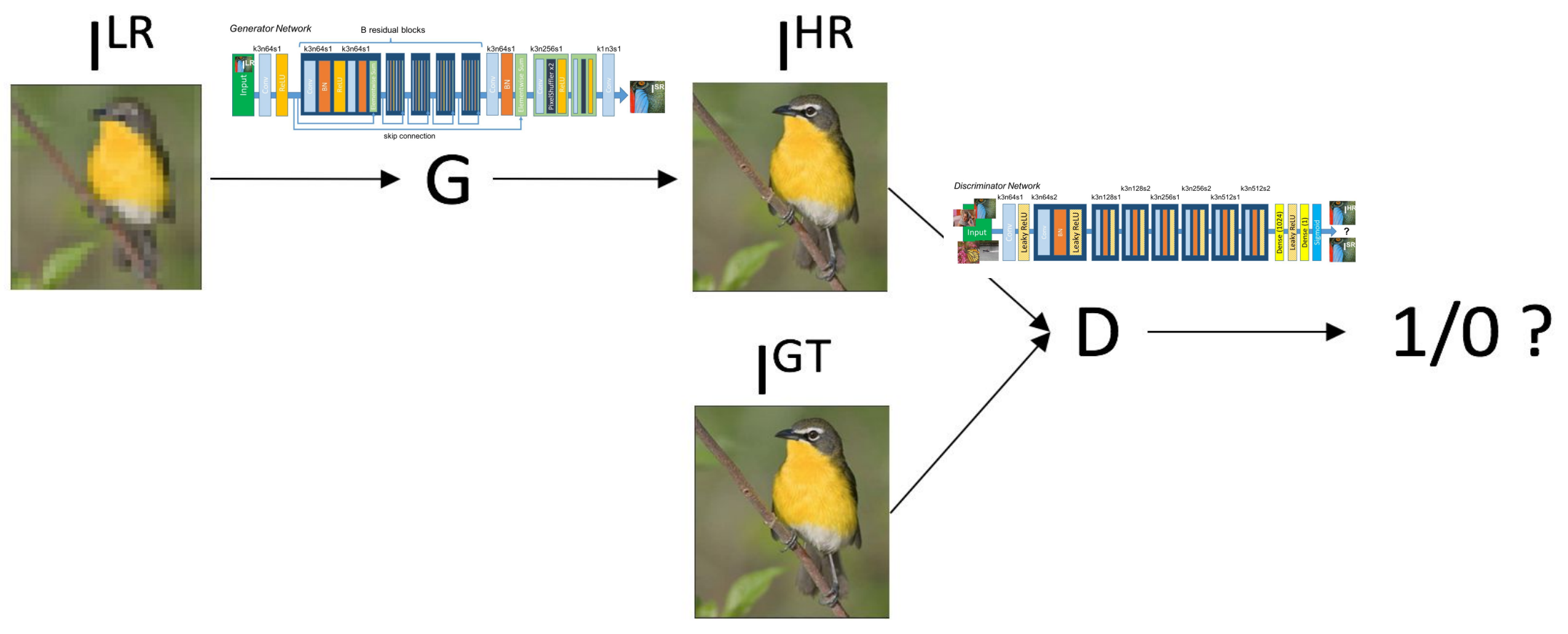
$$l_G^{GT} = -\sum_{n=1}^{N} logD_{\theta_D}(I^{GT})$$

$$loss = l_G^{SR} + l_G^{GT}$$

Training Detail:

According to the min-max loss function, we first fix generator G, update the weights in discriminator D twice, both for the generated images and ground truth images. After updating the weights in D, we then fix the weights in D, update the weights in G according to the perceptual loss function.

Initialization of G: we used the weights after 100 epoch training from the Deep Convolutional network to avoid convergence to local optimal.



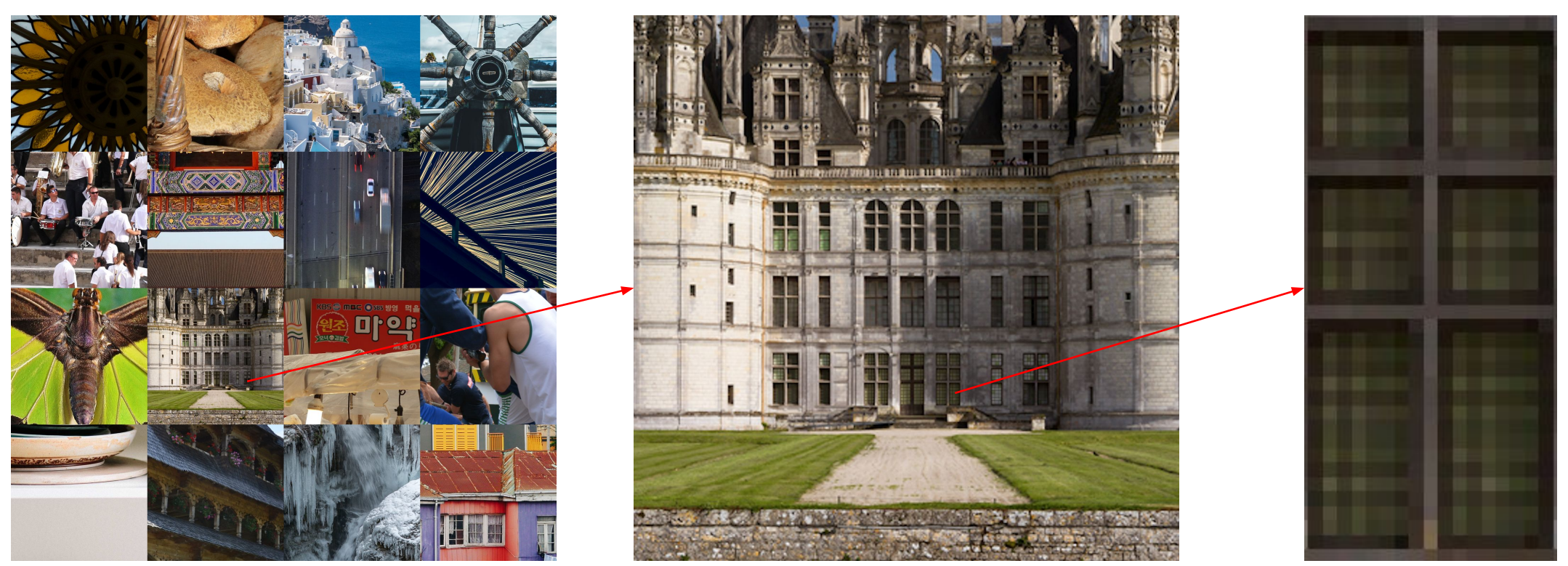
Batch size: 32
Adam Optimizer with BETA1: 0.9
Learning rate: 1x10^{-4}, decaying by factor 0.1 after 300

epoch
Input image size: 56 x 56 x 3
Output image size: 224 x 224 x 3
Epochs of training: 630

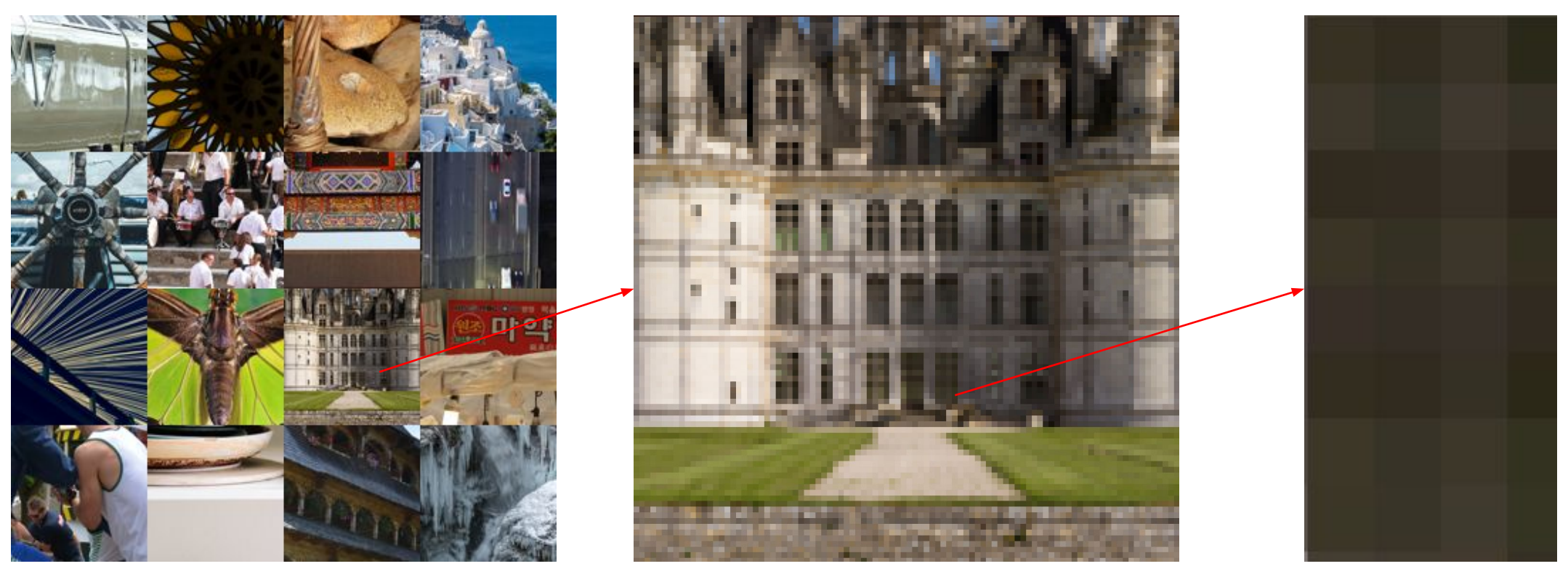
The tricky point in the training is for the perceptual loss, we have to rescale the vgg loss to be comparable to the mse and adversarial, those are based on the paper[1]

Result

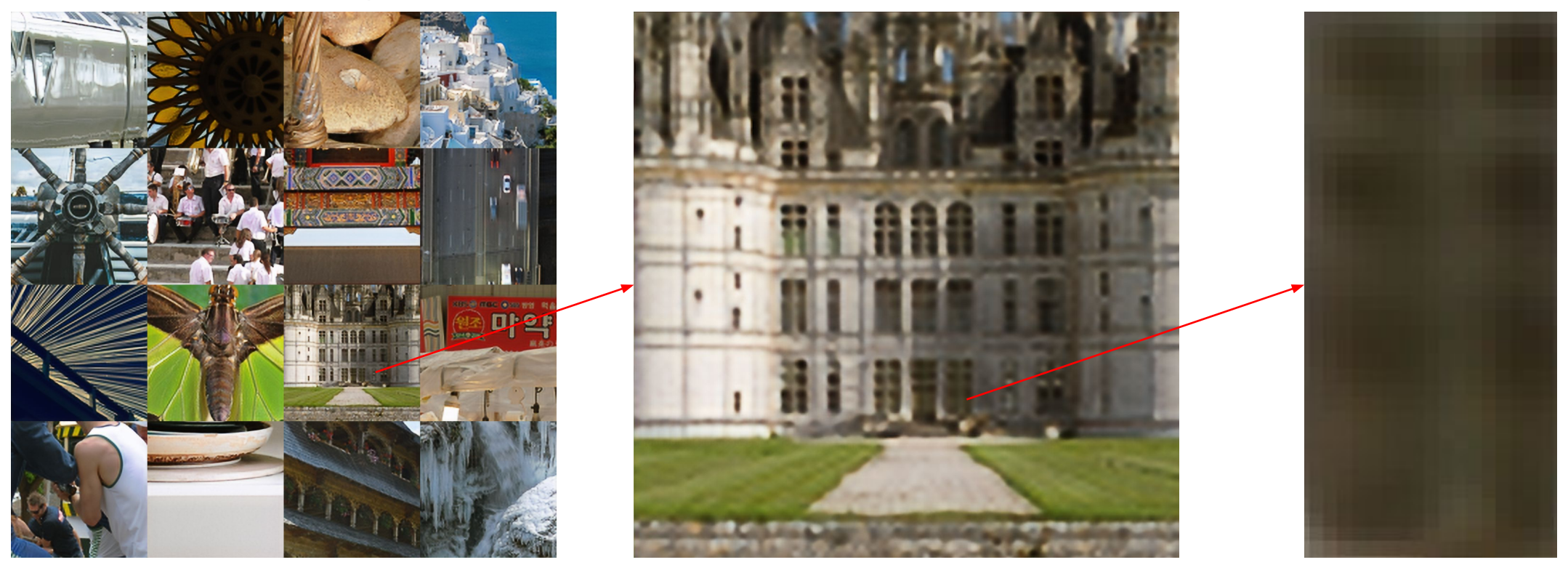
Ground truth:



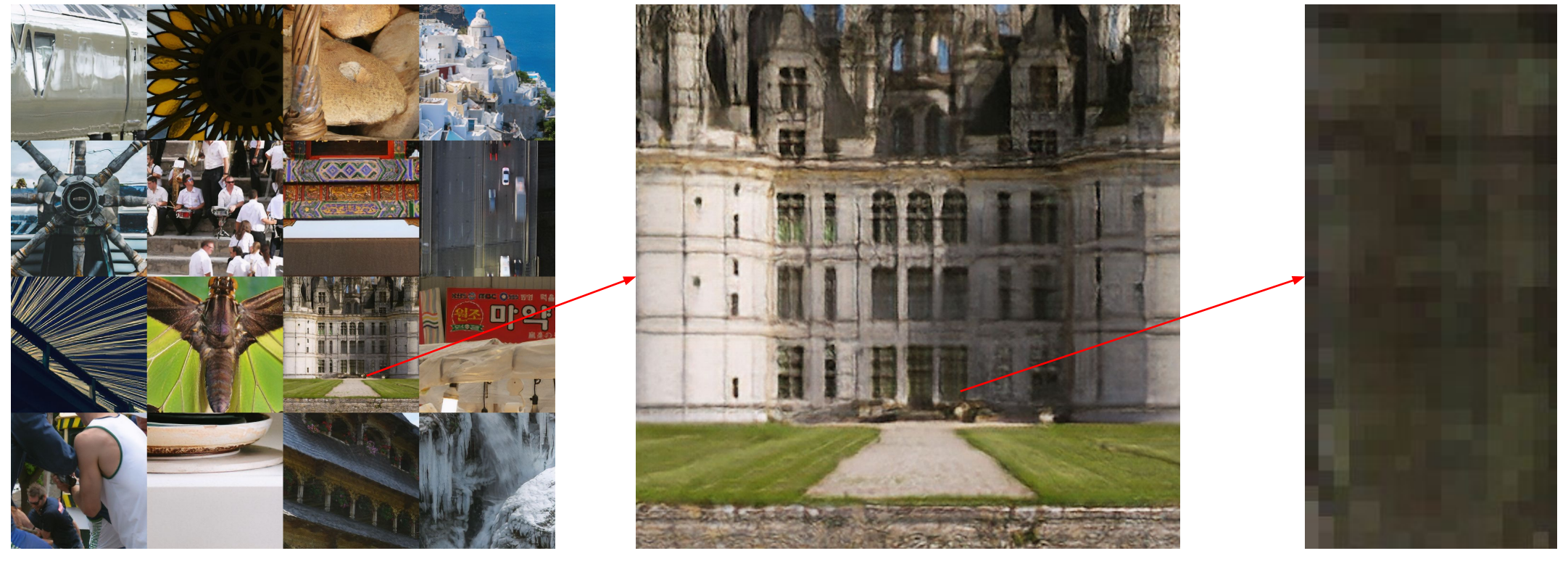
Input images:



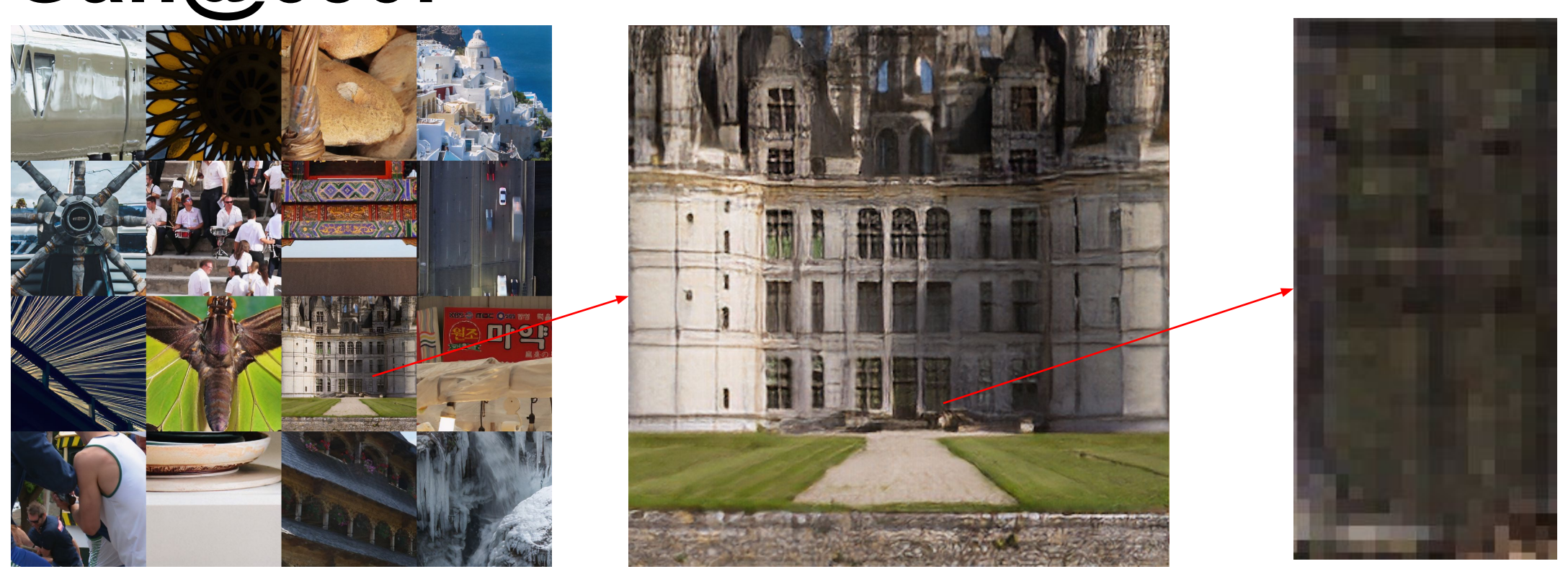
MSE only:



Gan@300:



Gan@630:



As shown above, comparing with the model with only mean squared error loss, our project works better. And with more epochs of training, the result turns better.

Reference

[1] Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *arXiv preprint arXiv:1609.04802* (2016).
[2] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
[3] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.