Image Super Resolution using Generative Adversarial Networks

Yi-Heng Huang and Yi-Wen Wang

**Abstract**—Since computer was invented, scientists hope to create artificial intelligence stuff through the efficiency of computers. However, the computer performance is too low to get this target, and it’s not have enough data for train, but with the semiconductor technology improvement and data collection, the artificial intelligence gets attention again, neural networks is widely used in many fields. We proposed a new architecture base on Generative Adversarial Network, and adding Structural Similarity as an loss function, and using Wasserstein distance replace Kullback-Leibler distance to improve the capability of discriminator, to generate a better high resolution image.

**Index Terms**—Convolution Neural Network, Machine Learning, Super-resolution

—————————— ◆ ——————————

# 1 Introduction

T

he image resolution depends on the characteristics of the lens and the sensor In the aspect of hardware. due to the algorithm improvement, super resolution algorithm can enhance the low-resolution image into a better resolution one. It’s divided into two parts: (1) reference from internal pixels to generate higher resolution image details (Internal Sample), such as the original interpolation. (2) reference to multiple external images (External Sample), which produces details corresponding to the texture contour of the original image, for example, a convolutional neural network to train the correlation between multiple original images and high-resolution images. After the training is completed, network can generate the detail pixel. like the previous works Super Resolution Convolutional Neural Network (SRCNN) [1] and the recently work Super Resolution Using Generative Adversarial Network (SRGAN) [2].

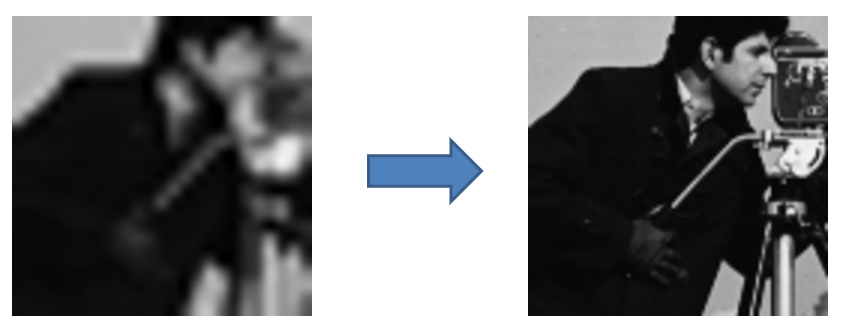
Training network with the low resolution and high resolution of image data for learning the correlation. The network architecture is based on SRGAN. It is mainly divided into a generator and a discriminator. The goal of the generator is to generate a high-resolution image from a low-resolution image. The discriminator is to distinguish of the image is fake who generates from generator or real data. 

Fig. 1 Illustration of Super-resolution

# 2 Related Work

In this chapter, we will introduce the Convolutional Neural Network (CNN) and the Super Resolution Using Generative Adversarial Network (SRGAN).

## 2.1 Convolutional Neural Networks

The Convolutional Neural Network (CNN) [5], because of the particularity of convolutional layer, it is very helpful in finding the correlation in the image space, and the correlation between individual pixels only appears nearby. pixels have low correlation with far one, so there is no need to deal with image problems in a fully connected layer, which can greatly reduce the time complexity and space complexity, so it is widely used in image processing fields.

CNN basely have two parts, (1) the feature extractor, (2) classification. Feature extractor composes of many convolutional layers, it’s mainly used to extract the features from input image. Classification is usually a fully-connection network, which are used to correlate or classify previously extracted features. It can also be used as a data representation, like the basis in linear algebra, the concept of sparse coding.

## 2.1.1 Convolutional layer

Convolutional Layer, the main concept is using Kernel to process images like the well-known image processing, but it’s often executing with overlap in the sliding window. The purpose is to avoid loss information. However, in fact, the size of the sliding window (Stride) is still adjusted according to the practice. Figure 2.2 shows the action diagram of the convolutional layer. The input is 7 x 7 image (I), the kernel (K) of the operation is 3 x 3, and the sliding window of step is 1, for the image area covered by the kernel, and makes a dot product (Matrix Element Wise Multiplication), the output is a 5 x 5 feature map (I\*K).

Each value in the kernel is the weight to be learned by networks. Filters is composed by many kernels. They learned the correlation between input and output, the product called feature maps.

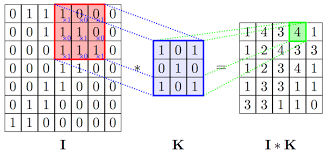


Fig. 2 Illustration of Convolution

## 2.1.2 Pooling Layer

The purpose of the pooling layer is mainly to reduce data (Down-sampling), retain most of the more important information, and reduce computation effective. The disadvantage of this operation is that some information may be lost, so it must be used properly. It can also be used to adjust the output and input dimension coupling between layers. Pooling operations can generally be aspect of two types: (1) Max Pooling and (2) Mean Pooling. The operation method is also to use the pooling kernel and image intersection area to select the area. The internal maximum value is used as a representative value or an operation for calculating the average value of each element in the region. The detailed operation method is input as a 4 x 4 matrix as shown in Fig. 2.3, the pooled kernel is 2 x 2, and the step is 2.

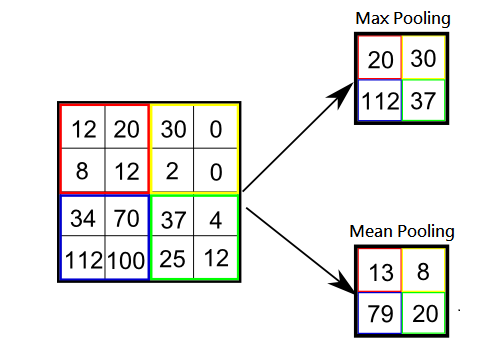


Fig. 3 Illustration of Pooling layer

## 2.1.3 Activation Function

When feedforwarding step, the input is multiplied by the weight and added to the bias. the activation function must be used to normalize the output value, and because the previous operation belongs to the linear function conversion. The problem function to be required is very complicated. When the conversion of the linear function is no longer satisfied, the activation function plays an important role. It can make the network have a better mapping ability for complex problems. The commonly used activation functions are The following S functions (Sigmoid function) have an output range of [0,1], Hyperbolic tangent function, output range between [-1,1], and linear rectification function (Rectified Linear) Unit, ReLU) is the input value is less than 0, the output is 0; when the input value is greater than 0, the output is the input value. And the deformation of ReLU, with Leaky Rectified Linear Unit (Leaky ReLU), does not completely filter out the input of the negative value, it’s a function with positive and negative (bipolar) output values.

every kind of activation function has preferred situation for use. For example, ReLU has a better convergence capability for deeper neural networks, and there is no problem that the gradient disappears. The S function and the hyperbolic tangent function is for network who using analog circuit design. The output value field has a positive and negative (bipolar) activation function because there is a positive and negative, there is a penalty mechanism for negative values. Compared with the activation function with only positive output, the network has a faster convergence ability, and it can reduce the training. time.

## 2.2 Generative Adversarial Network

Generative Adversarial Network (GAN) [3] is a network architecture that has been heatedly discussed and studied in recent years. Its special feature is that it consists of two neural networks, namely the Generator. The other is the Discriminator. In the case of the traditional neural network, it is usually to learn one-to-one or many-to-one mapping, and generate a probability distribution model that is a learning target against the network. There are characteristics of one-to-many mapping that are not available in general functions. This trait is closer to the intelligent biological mechanism.

In the beginning, generator randomly generates some input. Through the random input z, the generator G generates some output G(z), and then combine with real data (Ground Truth). given the correspond with the label(fake or real), then input to the discriminator, and the training for discriminating the authenticity. We hope generator can generate a fake data who can spoof the discriminator, and discriminator must learn how to distinguish fake or real. In the ideal state, G and D form D. (G(Z))=0.5, it has good capability when the same probability for distinguish it to true and false. x represents real data, z represents random parameters of the input generation network, E represents the event expectation value, Loss is calculated as follows, E\_(x~p(x))log(1-D(x)) indicates that the network must be extremely large for discriminating The two parts of this formula are: (1) discriminate that the network is true to the real data, and (2) E\_(z~p(z))log(1-D(G(z)) is generated by the network. The data is discriminated as a false expected value. Conversely, the generation network must be minimized (2). After the training is completed, the generation network is a false data that can randomly produce the distribution we want to be close to the real data. The optimization function is as follows:

## 2.3 Super Resolution using Generative Adversarial Network

The basic principle of Super Resolution Using Generative Adversarial Network (SRGAN) is similar with GAN. It is divided into generating network (Generator, G) and discriminating network (Discriminator, D). It is our purpose to generate network. Input and output are the low-resolution and high-resolution mapping relationship of the learning image. It can also be said to be the distribution model of the learning image enhancement method. The discriminator uses the judgment generator to generate the authenticity of the image, by using the enhanced discriminator. The ability to assist the generator to adjust so that the generator produces better high-resolution images.

In the aspect of generation of the network, the loss consists of two parts: (1) generator content loss, and (2) generator loss from discriminator. there are two mechanisms for loss in generator content loss, one of which is the mean square error of the high-resolution image (Super Resolution, SR) and the real image (Ground Truth, GT) produced by G (Mean Squared Error, MSE The other part is to send the SR and GT to the pre-trained, and remove the subsequent fully connected layer CNN architecture VGG19[4] model, and the SR feature map and GT feature map generated by the final layers of the convolution layer Then use MSE to calculate Loss.

To determine the loss of the network part, the binary code error is calculated by simply calculating the error between SR and GT and its label.

## 2.4 Residual Network

The residual network is composed of multiple Residual blocks (as shown in the figure). The main concept is derived from statistics. Here is the residual that is intended to make the network learn input and output, reduce some networks. Learning load, reducing the width of the network. The block contains multiple neural network layers, usually requires more than two layers and contains nonlinear activation functions, so that a single block will have better nonlinear mapping capability, otherwise the product of the input and the weight alone will be added. The bias is just a linear function and the mapping ability for the input and output relationships of the problem is also poor.

The residual block represents the figure. x is the input, F(x) is the output of the mapping function representing this block, and x identity is the shortcut. It can be thought of as bypass if you do not need to use certain layers when training the mapping function. Drop some layers, or you can adjust how much upper-level output information is needed, mix it with the output of this layer, retain some previous information, and send more information like post-transmission, which is good for feature extraction or regression.

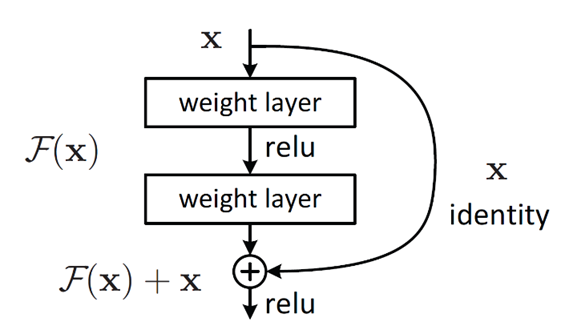


Fig. 4 Illustration of Residual block

## 2.5 Wasserstein GAN

For improve the GAN, the problem of gradient vanishing caused by the discriminator training is good, so that the loss from discriminator to the generator is reduce or even disappeared to cause the generator became weakened. The traditional GAN is to minimize the KL distance between the data generated by the generator and the actual data (Kullback-Leibler Divergence), and must maximize ability of the discriminator, and must minimize the distance between the generator and the real data, it’s like a gamble. Wasserstein GAN [6] made some adjustments here, importing the Wasserstein Distance, and the discriminator becomes used to score the input image. Weight clipping is used to limit the ability of the discriminator and to eliminate the training of the discriminator initialization. Make the discriminator not because the ability is too strong, the gradient disappears, and the loss cannot be passed to the generator.

## 2.6 Gradient Penalty

The problem of weight clipping is that the range of weight limit is very narrow, the range is too large to cause gradient explosion, too small will lead to gradient vanishing, and the weight limit will lead to the distribution of weight values. At the two boundaries of the limit value, the discriminator is degenerated into a binary function, which greatly reduces the discriminator's ability, and the auxiliary generator's training effect is poor. Therefore, Ishaan Gulrajan et al. Improved Training of Wasserstein GANs (Wasserstein GAN GP) [7]. The gradient penalty in the text improves the problem caused by weight clipping by limiting the gradient. The main concept is derived from Lipschitz continuity, |f(x\_1 )-f(x\_2 )|≤K|x\_1-x\_2 |→ReLU[〖||∇\_x D(x)||〗\_p- K], the gradient is limited to not exceed K, so that the weight value has a more normal distribution, as shown in Figure 5. Improve the problem of weakening the discriminator.

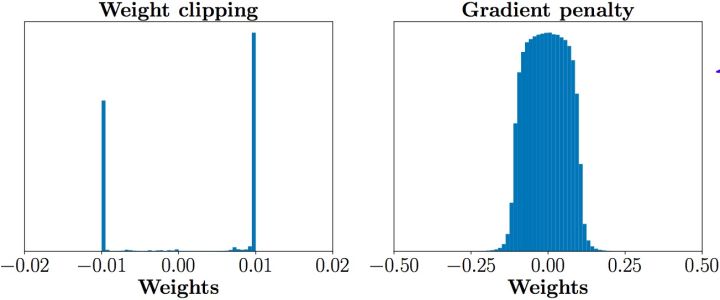


Fig. 5 the different between with Gradient Penalty or not

## 2.7 Structural Similarity

The index used to measure the similarity of two digital images. When two images are distorted images and undistorted images, the quality of the distorted image can be measured through structural similarity.

Image data is highly structured, and there is a strong correlation between pixels. Such associations contain structural information of objects. Human vision has become accustomed to extracting structural information naturally, so structural similarity (Structural Similarity) [8] is more in line with the human eye's intuition of image quality than the Peak Signal to Noise Ratio (PSNR). The given images x and y represent the matrix of pixel values of the two images, respectively. The structural similarity between the two is defined as the following formula: l(x, y) represents the luminance values of the x and y pixels, c(x , y) represents the contrast of the two, s (x, y) represents the structure of the two, α, β, γ are used to adjust the important parameters of l, c, s, the value must be greater than 0; , is the pixel mean value , of x and y, respectively, which is the pixel standard deviation of x and y, respectively. The formula is as follows:

## 2.8 Multi-scale Structural Similarity

Multi-scale Structural Similarity [9], commonly used in image evaluation methods, is an extension of structural similarity, using multiple distortion signals of different scales, multiple different The true signal of the scale, the structural similarity algorithm, gives different weights to each scale, and multiple scales together give the degree of similarity of the structure. The algorithm is shown in Figure 6.

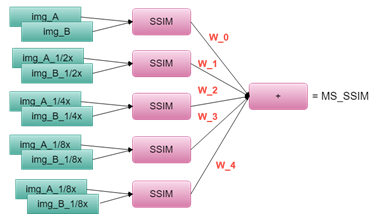


Fig. 6 Multi-scale Structural Similarity

## 2.9 Resize Convolutional Layer

Resize convolutional layer is often used to replace the deconvolution layer, mainly because the image reconstructed by deconvolution is easy to have a checkerboard artifact as shown in Figure 7. This phenomenon is that the reconstructed image will appear more and more clearly. , similar to the checkerboard pattern. The scaled convolution layer basically uses the interpolation method to resize the input image or the feature map, instead of the un-pooling layer of the deconvolution layer, and then uses the convolution to improve image quality.

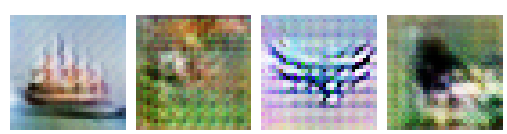


Fig. 7 Checkerboard artifact

# 3 The Proposal Algorithm

Super Resolution Using Generator Adversarial Network (SRGAN) is the application of GAN in super-resolution. This article is based on this architecture and uses structural similarity (SSIM) in content loss. Multiscale SSIM replaces the original Mean Square Error (MSE). And introduce Wasserstein distance, replace the original KL distance, improve the double game phenomenon, and add gradient punishment to improve the defects of weight trimming. After the training is completed, the part of the network generated by SRGAN can learn the mapping relationship between low-order resolution and high-resolution.

## 3.1 Architecture

This work follows the architecture of SRGAN, and makes some modifications and adjustments to the generated network and loss function parts, and cross-checks the effects of each combination, and analyzes various reasons from the results. The architecture is mainly divided into three parts: (1) Feature extraction and representation: This part is used to extract the low-resolution feature map as the input of the objective function. (2) Non-linear mapping: a non-linear mapping between a low-resolution feature map and high resolution. (3) Reconstruction: A feature obtained by low resolution is subjected to non-linear mapping to learn a high-resolution pixel block, to reconstruct a high-resolution image.

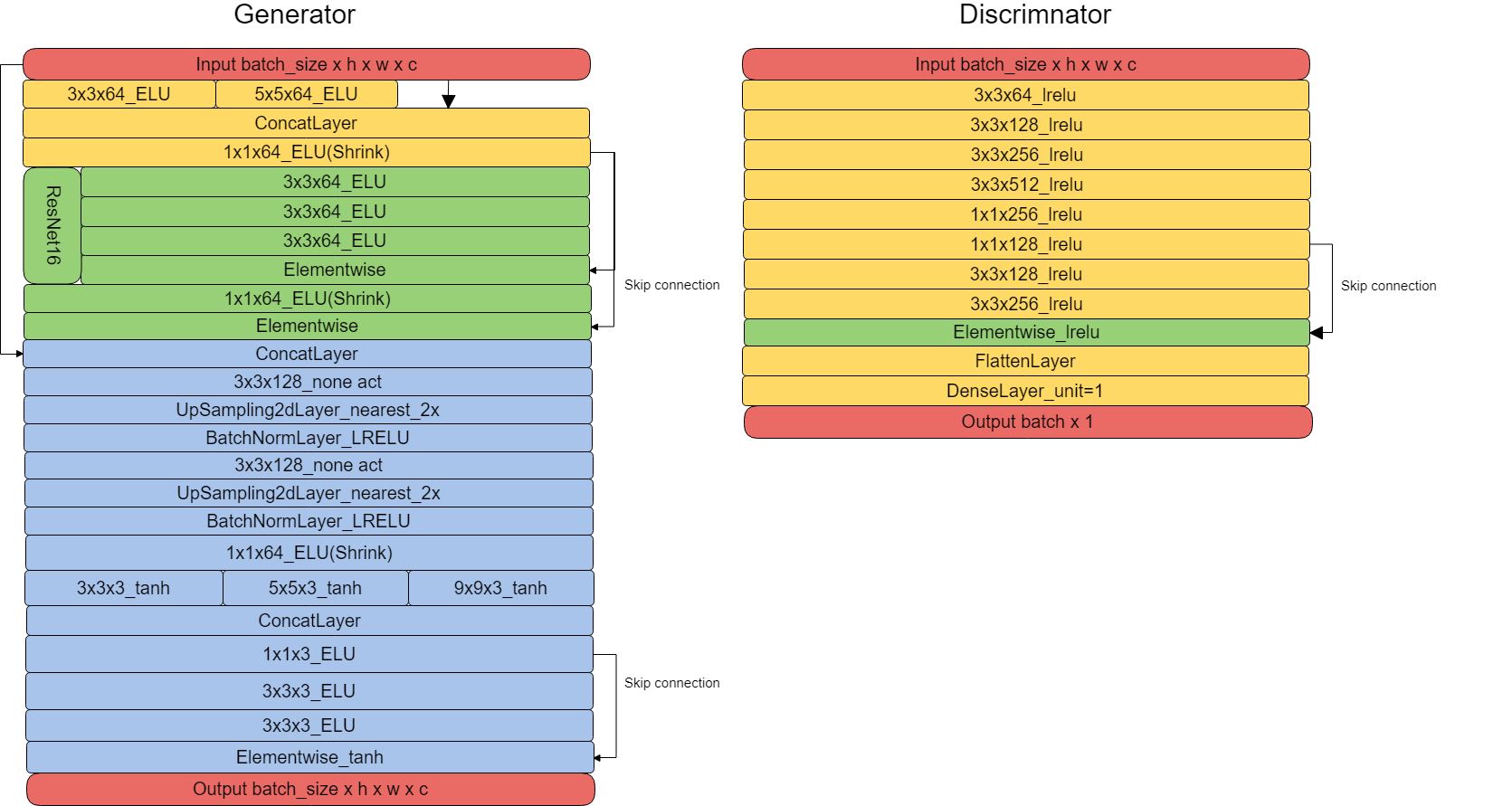


Fig. 8 Generator

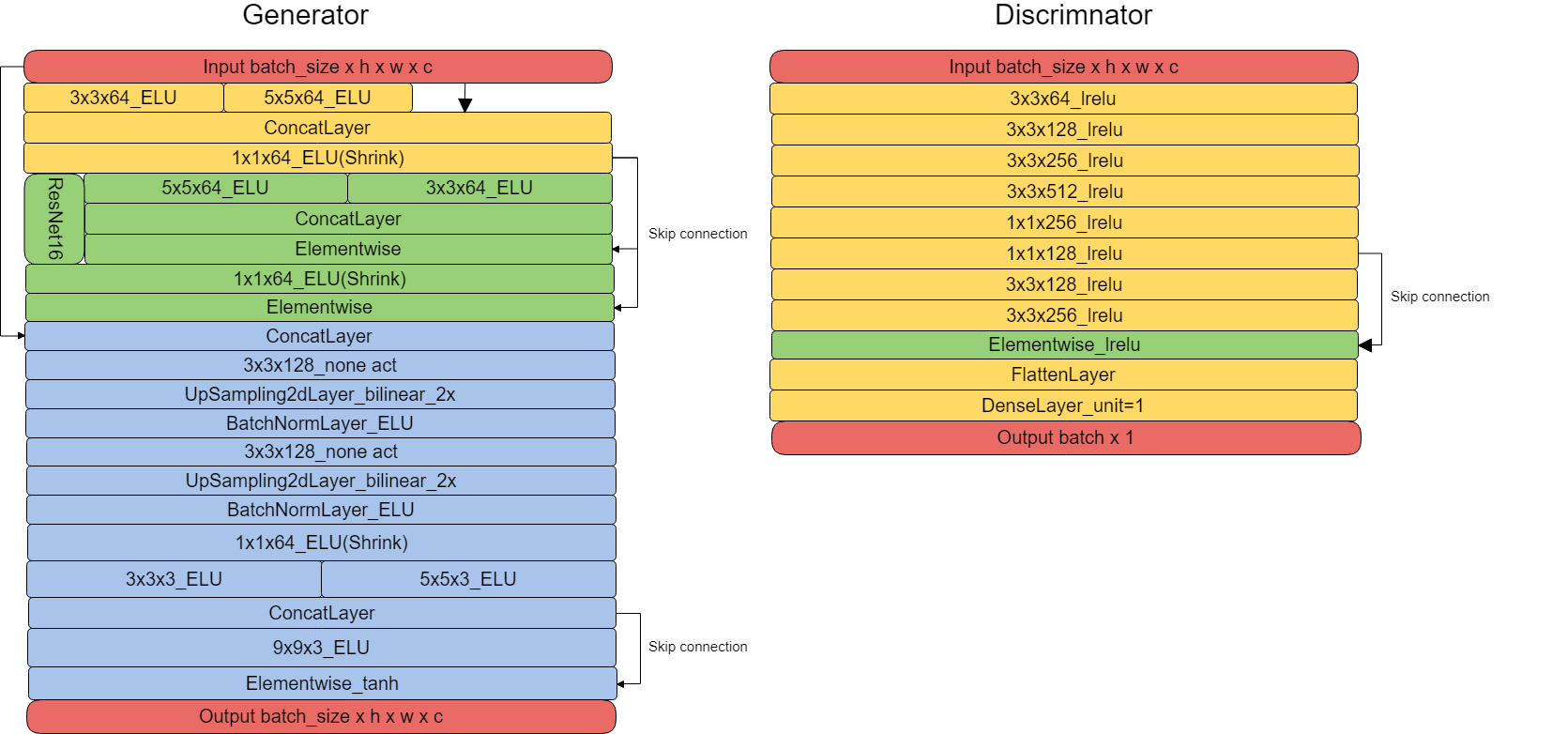


Fig. 9 Discriminator

## 3.1.1 Feature extraction and representation

This paper extracts the feature representation, using two parallel convolutional layers, using 3x3 and 5x5 convolution kernels respectively. The main reason is that it is possible to obtain richer features with different Receptive fields. The first two acquired features are stacked and a convolutional layer with a 1x1 convolution kernel is used to compress these features, on the one hand to increase the feature concentration and reduce computation.

## 3.1.2 Non-linear mapping

This section uses a 16-block residual network. The main idea is to try to let the residual block learn the mapping needed to generate high-frequency signals, and combine the high and low frequency signal features to generate high-resolution images. Feature blocks facilitate rebuilding high-resolution images. Two sizes of 3x3 and 5x5 convolution kernels are used as feature maps for different visual receptive fields.

## 3.1.3 Reconstruction

SRGAN reconstruction part uses Pixel-shuffle Convolution to reconstruct high-resolution images. This article will use the Resize Convolution instead of the original reconstruction method.

## 3.1.4 Loss Function

Mean Square Error, Structural Similarity, and Multi-scale Structural Similarity are used as content loss.

## 3.1.5 Experiment Lists

Loss is based on the Backpropagation algorithm, which passes forward to the previous number of layers and updates the weight of each layer when training the network. Divide into the following paths:

(1) G\_loss\_VGG: indicates that the pre-trained VGG19 loss function is used to correct the generator weight.

(2) G\_loss\_content: indicates the correction of the generator weight from the mean square error or structural similarity loss function.

(3) D\_loss: indicates the correction of the discriminator weight.

(4) G\_loss\_from\_D: Correction of the generator by the discriminator trained for a period of time.

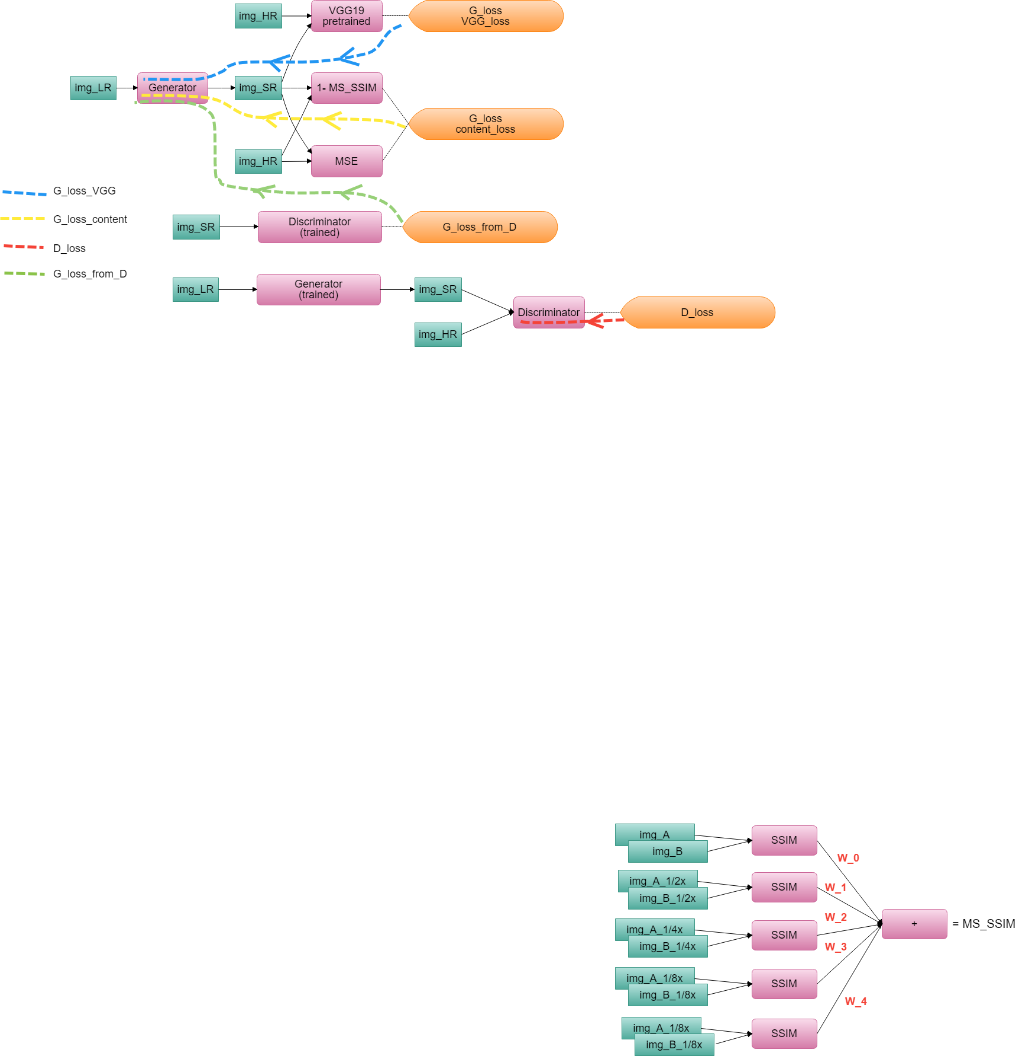


Fig. 10 Loss Backpropagation Flow

## 3.2 Setting

## 3.2.1 Dataset

Using the STL10 data set provided by Stanford University, select some of the truck and boat data as training and test data. The training set contains 896 images, and the evaluation and test set share 16 images. The original size of the STL10 data set is 96x96. On the 96x96 image, we randomly take out 48x48 high-resolution images as Ground truth, and then make 48x48 real data into Bi-cubic. Reduce the algorithm to 12x12 as the input image for low-resolution.

## 3.2.2 Parameters and Input /Output

(1). Training image input size (input size): 12x12

(2). Training image output size (output size): 48x48

(3). Batch training size (batch size): 16

(4). Learning rate:

(5). Learning rate decay rate:

(6). Learning rate decay epoch: 2500

(7). Training period (training epoch) : 5000

# 4 Experiment

## 4.1 Experiment of Environment

The environment setting introduction, the CPU is the Intel Core i7 CPU 4.70-GHz, the operating system is 64-bit Windows 10, and the implementation platform is Tensorflow version 0.12.0-rc1.

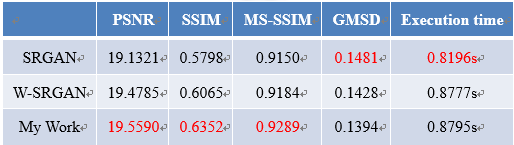
## 4.2 Results and Analysis

## 4.2.1 Comparing SRGAN, Wasserstein-SRGAN, and My Work

Comparing SRGAN, Wasserstein-SRGAN, and the method in this paper, the common image quality assessment peak signal to noise ratio (PSNR), structural similarity (SSIM), Multi-scale structural similarity (MS-SSIM), gradient magnitude similarity deviation (GMSD), to measure image quality. The results are shown in Table 1.

Table 1

**Comparing SRGAN, Wasserstein-SRGAN, and the method in this paper**



This paper mainly compares the structural similarity between the generated image and the original high-resolution image, and the multi-scale structural similarity, mainly because the two measured image quality standards are closer to humans. The natural perception of the visual system. In both of image quality assessment, this article is better than the previous work, but because of the slightly larger architecture, the execution speed is slightly behind. Because the structural similarity series is used to calculate the content loss, the dependence on VGG loss is large. Under the same training period, the convergence speed is slower than the mean square error. If there is more time training, the effect should be better. The mean square error method has a good network convergence when the number of cycles is about 1000 cycles, and the loss has been reduced to a very low level. From this point of view, the architecture of this paper has more potential. The result is shown in Figure 11.



Fig. 11 results

## 4.2.2 Comparing with Reconstruction methods

In this paper, we used two reconstruction methods for comparison, namely Pixel-shuffle Convolution and Resize Convolution. The results are shown in Table 2.

Table 2

**Comparing with Reconstruction methods** 

Resize convolution in the reconstructed image is slightly better than the pixel shuffling convolution. The conjecture here may be that the number of convolution kernels in the previous layer is insufficient, and it is impossible to form enough feature fragments to reconstruct the pixel convolution, and the scaling convolution is It is very intuitive to enlarge the feature map and then adjust it. However, the execution time is better for pixel convolution, because the feature fragment data reshapes, which must be faster than the interpolation feature map. The result is shown in Figure 12.



Fig. 12 results

# 5 Conclusion

Structural similarity is closer with human visual perception. This method has the value of extension, but the architecture of this paper not good enough. The loss function using pre-trained VGG19 is smart method, but we are not sure whether there is a dependency between this method and the mean square error loss function or the structural similarity loss function. As a result, during training, there may be adverse effects on network convergence and network capability. We think that the first layer of the convolutional layer pre-trained VGG19 can be added to the structure of this paper. The feature extraction and representation part enhance the detailed feature map, so that the subsequent network has better mapping and Refactoring ability, and removing the original VGG19 loss function, reducing the impact of the dependence of the loss function. The structure is shown in Figure 13.

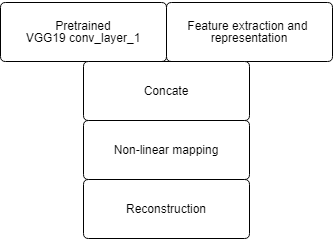


Fig. 13 Future Work

**References**

[1] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang."Image Super-Resolution Using Deep Convolutional Networks" Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)， 2015.

[2] 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 (CVPR), 2017.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. “Generative Adversarial Networks” 2014

[4] Karen Simonyan, Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition” Pattern Recognition (ACPR), 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun “Deep Residual Learning for Image Recognition” Computer Vision and Pattern Recognition, 2015.

[6] Martin Arjovsky, Soumith Chintala, Léon Bottou. “Wasserstein GAN” arXiv.org Machine Learning (stat.ML); Machine Learning (cs.LG) 2017.

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville ” Improved Training of Wasserstein GANs ” arXiv.org Machine Learning (cs.LG); Machine Learning (stat.ML) 2017.

[8] Zhou Wang, Member, Alan Conrad Bovik, Fellow, Hamid Rahim Sheikh, Student Member, and Eero P. Simoncelli. ”Image Quality Assessment: From Error Visibility to Structural Similarity” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600−612, Apr. 2004.

[9] Zhou Wang, Eero P. Simoncelli and Alan C. Bovik. “Multi-scale Structural Similarity for Image Quality Assessment” The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003.Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.pp, Issue.99 , 2016.