Image Super-Resolution Using Super-Resolution Convolutional Neural Network

Wensong Zhang & Tiancheng Sun

384 Zijinshan Street, Huangdao, Qingdao, Shandong, 266555, China.

E-mail: [zhangwensong@vip.qq.com](mailto:zhangwensong@vip.qq.com)

**Abstract**

Image Super resolution (SR) can convert a smaller image to a high-resolution image. Because different prior knowledge can lead to different result. So, there is no denying that its result cannot be unitue, absolutely. Super-Resolution Convolutional Neural Network (SRCNN) is an example-based method to achieve the SR. It learns the prior knowledge first, then build model based on these studies and predict the output. We implement this method by taking the original image, using bicubic interpolation convert it to the target size, then doing the non-linear projection based on seven layers CNN mapping between low and high resolution images. Finally, output the high-resolution image. CNN works as Picking image pixel and build sparse coding dictionary, image rebuild after projection. Our programming language is Python. We used Tensorflow and Scipy as the main packages, the mian dataset applied in trainging is DIV2K photo bank, and we used the Mean Square Error (MSE) to build the loss function and the Peak Signal to Noise Ratio (PSNR) to evaluate our result.

**Keywords**: Image Super resolution, Super-Resolution ConvolutionalNeural Network, TensorFlow

1. **Introduction**

Single image super-resolution, which aims at recovering a high-resolution image from a single low-resolution image, is a classical problem in computer science. Based on the mutability of the low-resolution image and the way to solve the problem, there are varies solutions exist. So, instead of saying it is a question, single image super-resolution is more about a kind of understanding, because its solution is not unique.

For this project, the model we used is called Super-Resolution Convolutional Neural Network. The reason why we choose this method is that, first, its structure is intentionally designed with simplicity when it was built up, and it provides higher accuracy compared with other methods which are created at almost same time. Second, with a large number of filters and layers, this method achieves faster speed for useful online usage even on CPU and GPU. This method is faster than a number of example-based methods, because it is entirely based on feed by and does not need to deal with any other complex problem on usage, which gives user more possibility when they are using this method. Third, there are some studies show that there is huge space and higher quality could be used for bigger and faster model when there are large and multiple data can be stored in to the network. Moreover, the network provided by this method givens a kind of possibility for the end-to-end mapping between low-resolution and high-resolution images, at the same time, there is not too much pre-requirements for the pre/post processing about this method. Also, there is a relationship between this method with the traditional methods, and it provides a guidance for the implementation the network structure of this method. It shows that the basic computer science problem can be solved in a fast and efficient way.

According the end-to-end mapping between low and high resolution images, CNN keep high efficiency and fast speed depending on its lightweight structure. Based on above advantages provided by CNN, we choose it as our algorithm to finish this project.

1. **Problem Statement**

One of the most difficult problem for us is how to implement the image super resolution using CNN algorithm. If we divide it in parts, there are basically three main parts.

First is Image Super Resolution. The traditional method focuses on how to learn a compact dictionary or manifold space to relate low/high-resolution patches and how to represent these schemas. They use low/high-resolution patch pairs to present the dictionary directly. They use the nearest neighbor of the input as the low-solution patch and the high-resolution patch is using for reconstruction. This ensure the patch extraction and aggregation steps is the most important part in optimization. They are called pre/post-processing and they should be executed sepreatedly.

Second is Convolutional Neural Networks, which is also called CNN. It has huge success in handle image classification, it provides a chance to get the efficient training implementation of GPU. As the same time, it gives much faster speed while maintaining the high quality. It also allows to access a huge numebr of data when training a complex model.

Last, the Deep Learning for Image Restoration. When the deep learning is working on image restoration, all the layer is fully connected, it is used for reducing image noise. At that time, the CNN is using for image denoising and removing noise pattern, while it is aiming to solve the image restoration problems.

1. **Literature Review**

Learning a Deep Convolutional Network for Image Super-Resolution: this is the famous SRCNN (Super-resolution Convolutional Neural Network) journal that first apply data mining techniques to the single image super resolution problem. Within this article, the author points out that the traditional super-resolution method can be replaced by the convonlutional neural network’s convolution operation and the activation function. Since this is the first article that apply CNN into super-resolution problem, the author only designed a simple structure which contains only three convonlution layer. The first layer respons for extraction the features in the low-resolution image (which has already been extended into the target size by a pre-processing block that is outside of the CNN) then feed it into the second layer. After received the result given by the first layer, the second layer will do a non-liner mapping to map the low-resolution images’ features into the high-resolution image’s feature. Then, the third layer will take this mapping and reconstruct the high-resolution output image.

Accelerating the Super-Resolution Convolutional Neural Network (FSRCNN): this project is based on the result of the SRCNN project and provided several important improvements to it. The improvements mainly show in two aspects. First, the author added one deconvolution layer to extend the size of the image, therefore, now we can feed the Neural Network with the low-resolution images directly. Since the FSRCNN does not need to perform the image resize pre-process outside of the Neural Network, it obtained a significantly improve on its time complexity. Second, they allow different training model sharing the same set of non-linear mapping layers. Thus if model with different upsampling factor models is needed, the researcher only need to fine-tune the deconvolution layer.

1. **Methods and Techniques**

When the SRCNN algorithm is working, it takes the low-resolution image as the input, after the patch extraction and representation we will get a n-feature maps of the low-resolution image. Then, by doing the non-linear mapping, there will be a n-feature maps of the high-resolution image. Finally, after the reconstruction is finished, we will get the high-resolution image as our output.

As one example in image-super resolution, our CNN model uses seven layers convolutional neural networks, but the quality of the final output is good enough. What has to be mentioned is that the image inputting to SRCNN is not the low-resolution image, but the image after applying bicubic on the original low-resolution image.

The style could be (conv1 + relu1) -> (conv2 + relu2) -> (conv3 + relu3) -> (conv4 + relu4) -> (conv5 + relu5) -> (conv6 + relu6) -> (conv7 + relu7), and and the size of each one are all 3\*3, the output features are 16, 32, 64, 128, 128, and 256.

We took the Y channel in YCbCr model, the detail process can be understood as that, inputting an image under BRG model, then convert it into YCbCr model, but only using the Y channel according the CNN network, then combine the output with the results getting by the other two channels. After that converting it back to BGR model. Last, take two BGR images and calculate the MSE loss, then do the Gradient Descent Optimization.

We tried one channel and three channels, but we find that one channel is better, there is lower chance to reach local optimum. Moreover, it would converge only if the learning rate of the last layer should be small enough.

1. **Discussion and Results**
   1. **Datasets**

Within this project, we mainly used the DIV2K data set from the “New Trends in Image Restoration and Enhancement workshop and challenge on image super-resolution” project. This project is a “call for papers” challenge that aiming at gathering image restoration and enhancement algorithm. In order to reduce the difficulty of gathering necessary data to finish this project, they made this dataset available on the internet for academic use. This dataset contains one high-resolution image's dataset and several low-resolution images dataset which produced by different downscale algorithms with different scale factors.

Within this project, we mainly used their high-resolution dataset during the training phase and the downscaling X3 database as our testing input. Originally, the training dataset contains 800 2K resolution high-quality images. All of these images are taken by the photographer with high-end cameras and stored as PNG format, therefore, these images contain no compression artifact and can be used as a representation of the target quality that we want our super-resolution algorithm to achieve from a low-resolution

image. However, due to the limitation of our computational ability, we failed to apply our train model on the whole original dataset. In order to obtain a meaningful model within the project processing period, we randomly selected 86 full-color images from the original dataset. This new training set contains 14 animal images, 9 portrait photo, and 63 landscape pictures.

Further, during the data preparation phase, we randomly cut the original input 2K resolution images into hundreds of small patches in order to make it suitable for our mining process since the TensorFlow is more capable for small parallel process rather than single highly complex one.

Although we are confident to announce that current dataset is the best dataset we can found and apply on this project in order to obtain an as universal as possible model for all kinds of image super-resolution work, it still has some obvious defect. The most obvious one is, the train data set is relatively small, so it failed to cover more image style like black-and-white photography and artwork. Thus, our trained model does not have a great performance while it takes these two kinds of images as input.

* 1. **Evaluation Metrics**

During the training and testing process, we mainly applied two evaluation metrics to evaluate the quality of our model, there are the mean square error and the Structural Similarity index.

Rebuild a high-resolution image base on a given low-resolution image means we want to minimize the difference between the rebuild picture with the original picture. Therefore, we can use the Mean Squared Error (MSE) algorithm as the loss function of the training process. Denote the original high-resolution image set used for training as {HR*i*}, the reconstructed picture set as {ReCi}, then we can express the MSE loss function as:

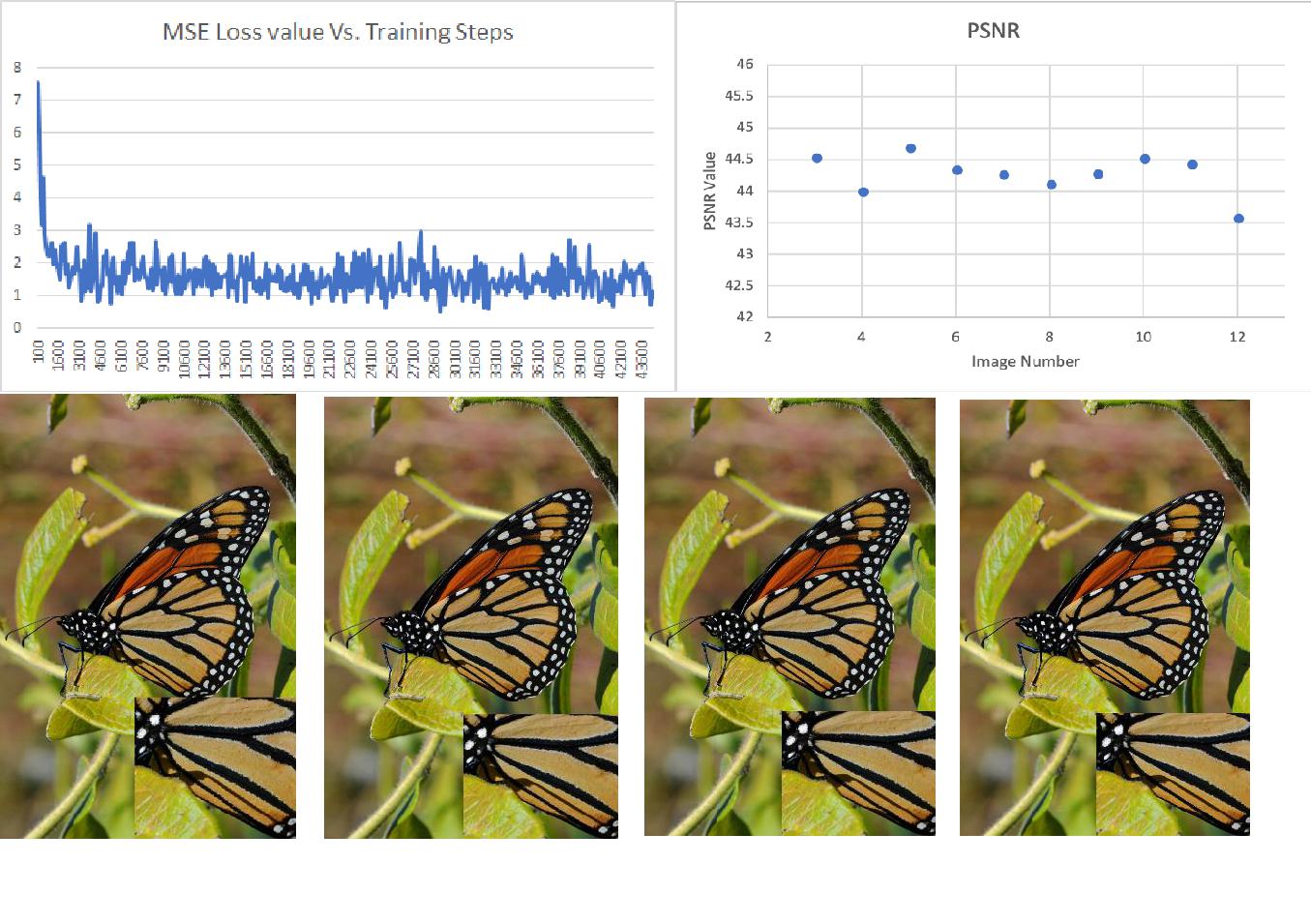
111

During the CNN training process, the value of LOSS will be reduced automatically through the neural network’s backpropagation.

Besides the MSE loss function that used within the convolutional neural network, we also applied the Structural Similarity Index as a supplemental of the MSE loss function.

Compare with the simple MSE loss function, the Structural Similarity index uses more information regarding the whole image to evaluate the similarity between two given picture, therefore, it is slower than the MSE loss function but can better represent the reconstructed image’s quality from human vision. We use this index to evaluate and optimize our training model after each training process.

* 1. **Experimental Results**



|  |  |  |  |
| --- | --- | --- | --- |
| INTER\_AREA | INTER\_CUBIC | INTER\_LANCZOS4 | Ours result |
| PSNR = 37.469 | PSNR = 38.347 | PSNR = 38.535 | PSNR = 44.377 |

After around 200k steps of training, our training model reached a higher PSNR value than current state-of-the-art super-resolution algorithm. Compare with interpolation method that used by OpenCV, our model shows a better result under the PSNR metrics. As shown in the PSNR chart, all of the 12 pictures’ PSNR value is above 43.5.

Although we have achieved a great result, we still have to accept the result that our model is still not steady. As shown in the MSE losss value Vs. training steps chart, the value given by the loss function is still not converge even after 40000 steps of training. We tried to solve this by increase the number of training steps to 100,000, but the loss value is still not steady.

1. **Conclusion**

At the end of this project, we explored several important super-resolution methods that based on Convolutional Neural Network and built a efficient training method for mining a good method to reconstruct a high-resolution image from a compressed artifact image. With limited training, our model has already achieved a performance that better than traditional super-resolution method that does not rely on the Convolutional Neural Network. We also comfired that current model’s performance can be improved by more training data and training steps. However, even our best model’s performance still significantly lower than the current cutting-edge super-resolution method that relied on the neural network if we evaluate them under the PSNR metrics. Further, our current model also failed to make use of the full-color information that contained in the image.

The main things we learned from this project is the possibility of applying data mining techniques that we learned from this course to the traditional computer science problem. Use this exercise as an example, although the super-resolution and image reconstruct problem has been raised for a long period and there are lots of complex algorithms, like bicubic and sparse-coding-based method, built to solve it. The new CNN method still gives a beautiful, light-weight solution. After noticing this possibility, we are able to apply the method we learned from this course to the traditional problems in other fields and try to find a new method just as what we did for the super-resolution project.

* 1. **Directions for Future Work**

In the near future, we are planning extend current random selected dataset to the full DIV2K dataset and observe the influence from the size of the training data to the performance of the model. Further, since we noticed the Recursive Neural Network and the residual network’s application during the literature review phase, we will be focusing on combining them with our current model and observe the performance. Since one of our team members will graduate this semester, he will respond for complete this project and execute the roadmap.

1. **References**

[1]Supplementary material: PSNR, SSIM, IFC, CORNIA results for top NTIRE 2017 challenge methods (SNU\_CVLab, HelloSR, Lab402), VDSR and A+ on DIV2K, Urban100, B100, Set14, Set5

[2]Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang. Learning a Deep Convolutional Network for Image Super-Resolution, in Proceedings of European Conference on Computer Vision (ECCV), 2014

[3]Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the Super-Resolution Convolutional Neural Network. Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 391-407. doi:10.1007/978-3-319-46475-6\_25

[4]Zhou Wang and Alan C. Bovik, "Mean squared error: Love it or leave it? - A new look at signal fidelity measures," IEEE Signal Processing Magazine, vol. 26, no. 1, pp 98−117, Jan. 2009.