Image Super Resolution using Convolution Neural Network

Aditya Lahoty

Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India

Abstract: The research paper has implemented a Deep Learning Model (Convolution Neural Network) for super resolution of a single image. The method directly learns an end-to-end mapping between the low and high resolution images. The model is basically a deep learning model which is using CNN in which input is a low-resolution image and output is the corresponding high-resolution images. The model has very lightweight structure, yet demonstrates state-of-the-art restoration quality, and achieves fast speed for practical on-line usage. We explore different network structures and parameter settings to achieve trade-offs between performance and speed. Moreover, we extend our network to cope with three color channels simultaneously, and show better overall reconstruction quality [1].

KEYWORDS: Super Resolution, Deep Learning, Computer Neural Network, Computer Vision

DOI of the Article: https://doi.org/10.46501/IJMTST0706028

Available online at: http://www.ijmtst.com/vol7issue06.html

As per UGC guidelines an electronic bar code is provided to seure your paper

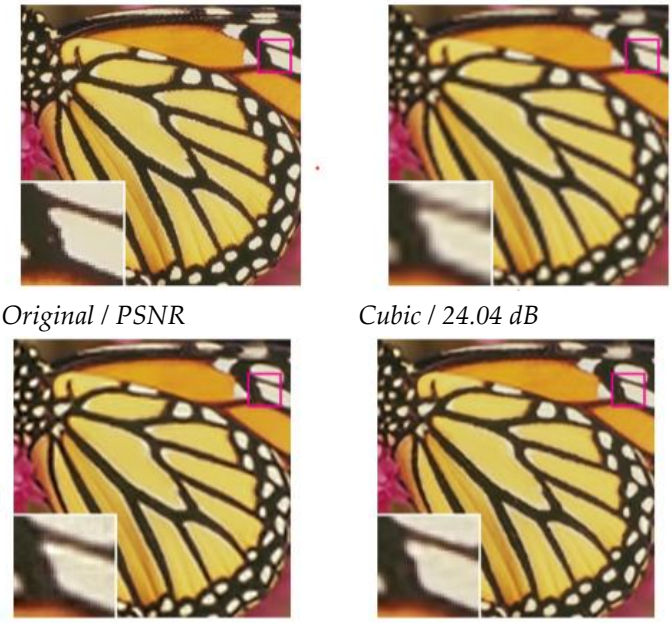
To Cite this Article: Aditya Lahoty. Image Super Resolution using Convolution Neural Network. *International Journal for Modern Trends in Sceicen and Technology* 2021, *7*, 0706100, pp. 162-168. https://doi.org/10.46501/IJMTST0706028

Article Info. Received: 16 May 2021; Accepted: 6 June 2021; Published: 13 June 2021

INTRODUCTION

The problem of getting a high-resolution [2] image from a low-resolution image is a very definitive problem in Computer Vision. This problem is inherently ill-posed since a multiplicity of solutions exists for any given low-resolution pixel. In other words, it is an underdetermined inverse problem, of which solution is not unique [1]. Such a problem is typically mitigated by constraining the solution space by strong prior information [1]. To begin, most models adopt the example-based methodology [3]. They change the interior characteristics of the same images or get functions from other low resolution/high resolution pairs. The external example-based methods can be formulated for generic image super-resolution, or can be designed to suit domain specific tasks, i.e., face hallucination [4], according to the training samples provided[1]. The sparse-coding-based method [5] is one of the representative external example-based SR methods [1]. This method has several points in its methodology. In the start, pre-processing of the image is done. The image formed is ciphered by a low-resolution function. The sparse coefficients are passed into a high-resolution dictionary for reconstructing high-resolution patches [1]. The overlapping re-constructed patches are aggregated (e.g., by weighted averaging) to produce the final output.

In this article, we prove the above pipeline is equivalent to a deep convolutional neural network [6]. Inspired by this fact, we considered a CNN, which can directly learn the end-to-end mapping between the low and high resolution image. Our method is fundamentally different from the existing methods based on external examples, because ours did not explicitly learn the dictionary which is used to model the patch space. These are implemented implicitly through the hidden layer. In addition, patch extraction and aggregation is also expressed as a convolutional layer, so it involves optimization. In our method, the entire SR pipeline is obtained completely through learning, with almost no pre-processing/post-processing. The proposed model has several appealing properties.



Original / PSNR *Cubic / 24.04 dB*

Sparse Method / 25.58 dB *CNN model / 27.95 dB*

Figure 1: The proposed model surpasses the bicubic baseline with just a few training iterations, and outperforms the sparse-coding-based method (SC) with moderate training. The performance may be further improved with more training iterations. [1]

First, its structure is intentionally designed with simplicity in mind, and yet provides superior accuracy compared with state-of-the-art example-based methods. Figure 1 shows a comparison on an example [1].

Second, with moderate numbers of filters and layers, our method achieves fast speed for practical on-line usage even on a CPU. Our method is faster than a number of example-based methods, becauseit is fully feed-forward and does not need to solve any optimization problem on usage [1].

Third, experiments show that the restoration quality of the network can be further improved when (i) larger and more diverse datasets are available, and/or (ii) a larger and deeper model is used [1].

On the contrary, larger datasets/models can present challenges for existing example-based methods. Overall, the contributions of this study are mainly in three aspects [1]:

1) We present a fully convolutional neural network for image super-resolution. The network directly learns an end-to-end mapping between low and high-resolution images, with little pre/post processing beyond the optimization [1].

2) We establish a relationship between our deep learning-based SR method and the traditional sparse-coding-based SR methods. This relationship

provides guidance for the design of the network structure. [1]

3) We demonstrate that deep learning is useful in the classical computer vision problem of super resolution, and can achieve good quality and speed [1].

The present work adds to the initial version in significant ways [1].

Firstly, we improve the SRCNN by introducing larger filter size in the non-linear mapping layer, and explore deeper structures by adding nonlinear mapping layers [1].

Thirdly, considerable new analyses and intuitive explanations are added to the initial results [1].

RELATED WORK

2.1 Image Super-Resolution

The super resolution algorithms are of four types - prediction models, edge based methods, image statistical methods and patch based methods. They are already tested and investigated in Yang et al.'s work [3]. The internal example-based methods decreased the images' similar characteristics and produces exemplar patches from the input image.

It is mentioned in Glasner's work [7], and many improved versions are put forward to speed up the work. The external example-based models train functionality between low and high resolution patches from external datasets. These theories differ on how to train a dictionary to work with low and high-resolution images, and on how representation schemes should be done in such work.

In the work of Freeman et al. [8], the dictionaries are directly presented as low/high-resolution patch pairs, and the nearest neighbour (NN) of the input patch is found in the low-resolution space, with its corresponding high-resolution patch used for reconstruction. Other mapping functions such as kernel regression, simple function, random forest and anchored neighbourhood regression are proposed to further improve the mapping accuracy and speed. [1]

Most of the sparse coding methods and its improvements [9], [10], [11] are among the state-of-the-art SR methods in present. In these methods, main optimization focus is the patches; the patch extraction and aggregation steps are considered as pre/post-processing and handled separately. The majority of SR algorithms focus only on one channel among the three channels. For color images, the aforementioned methods first transform the problem to a different space like YCbCr , and SR is applied only on the Y channel i.e., the luminous channel. There are also works attempting to super-resolve all channels simultaneously. For example, Kim and Kwon [12] and Dai et al. [13] apply their model to each RGB channel and combined them to produce the final results. However, none of them has analysed the SR performance of different channels, and the necessity of recovering all three channels [1].

2.2 Convolutional Neural Networks

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics [6].

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area [6].

2.3 Deep Learning for Image Restoration

Various studies have been performed for deep learning techniques for changing low resolution image into high resolution image. The multi-layer perceptron (MLP), whose all layers are fully-connected (in contrast to convolutional), is applied for natural image denoising [14]. More closely related to our work, the convolutional neural network is applied for natural image denoising [14] and removing noisy patterns (dirt/rain). These restoration problems are more or less denoising-driven [1].

Cui et al. [15] propose to embed auto-encoder networks in their super resolution pipeline under the notion internal example based approach [7]. The deep model is not specifically designed to be an end-to-end solution, since each layer of the cascade requires independent

optimization of the self-similarity search process and the auto-encoder. On the contrary, the proposed model optimizes an end-to-end mapping. Further, our model is faster at speed. It is not only a quantitatively superior method, but also a practically useful one [1].

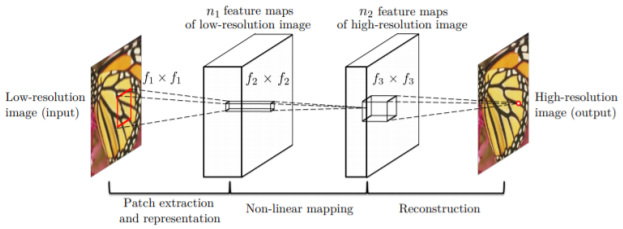


Figure 2: Given a low-resolution image Y, the first convolutional layer of the SRCNN extracts a set of feature maps. The second layer maps these feature maps nonlinearly to high-resolution patch representations. The last layer combines the predictions within a spatial neighbourhood to produce the final high-resolution image F(Y). [1]

PROPOSED METHODOLOGY

3.1 Formulation

Consider a single low-resolution image, we first pre-process it and upscale it to the desired size using bicubic interpolation, which is the only pre-processing we perform. Let us denote the interpolated image as Y. Our goal is to recover from Y an image F(Y) that is as perfect as the high resolution version X of the image. Y is the low-resolution image and X is the corresponding high resolution image. We have to train the model and make a function F, which conceptually consists of three operations:

1) Patch extraction and representation: This operation extracts patches from the low resolution image Y and represents each patch as a high-dimensional vector. These vectors comprise a set of feature maps, of which the number equals to the dimensionality of the vectors. [1]

2) Non-linear mapping: This operation nonlinearly maps each high-dimensional vector onto another high-dimensional vector. Each mapped vector is conceptually the representation of a high-resolution patch. These vectors comprise another set of feature maps. [1]

3) Reconstruction: This operation aggregates the above high-resolution patch-wise representations to generate the final high-resolution image. This image is expected to be similar to the ground truth X. We will show that all these operations form a convolutional neural network. A figure is depicted below regarding the methodology. Next we detail our definition of each operation. [1]

3.1.1 Patch extraction and representation

For image restoration, a strategy is to extract patches and then represent them by a set of pre-trained bases such as PCA, DCT, Haar, etc. This is equivalent to convolving the image by a set of filters, each of which is a basis. In our formulation, we involve the optimization of these bases into the optimization of the network [1].

Formally, our first layer is expressed as an operation F1: F1(Y) = max (0, W1 ∗ Y + B1), where W1 and B1 represent the filters and biases respectively, and '∗' denotes the convolution operation. Here, W1 corresponds to n1 filters of support c×f1 ×f1, where c is the number of channels in the input image, f1 is the spatial size of a filter. Intuitively, W1 applies n1 convolutions on the image, and each convolution has a kernel size c × f1 × f1. The output is composed of n1 feature maps. B1 is an n1-dimensional vector containing the biases, whose each element is associated with a filter. We apply the Rectified Linear Unit (ReLU, max (0, x)) on the filter responses [1].

3.1.2 Non-linear mapping

The first layer extracts an n1-dimensional feature for each patch. In the second operation, we map each of these n1-dimensional vectors into an n2-dimensional one. This is equivalent to applying n2 filters which have a trivial spatial support 1 × 1. This interpretation is only valid for 1×1 filters. But it is easy to generalize to larger filters like 3 × 3 or 5 × 5. In that case, the non-linear mapping is not on a patch of the input image; instead, it is on a 3 × 3 or 5 × 5 "patch" of the feature map. The operation of the second layer is [1]:

F2(Y) = max (0, W2 ∗ F1(Y) + B2).

Here W2 contains n2 filters of size n1 ×f2 ×f2, and B2 is n2-dimensional. Each of the output n2-dimensional vectors is conceptually a representation of a high-resolution patch that will be used for reconstruction. It is possible to add more convolutional layers to increase the non-linearity. But this can increase the complexity of the model (n2 × f2 × f2 × n2 parameters for one layer), and thus demands more training time [1].

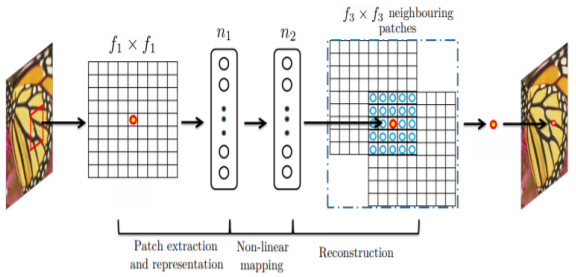


Figure 3: An illustration of sparse-coding-based methods in the view of a convolutional neural network.

3.1.3 Reconstruction

In the traditional methods, the predicted overlapping high-resolution patches are often averaged to produce the final full image. The averaging can be considered as a pre-defined filter on a set of feature maps (where each position is the "flattened" vector form of a high resolution patch). Motivated by this, we define a convolutional layer to produce the final high-resolution image: F(Y) = W3 ∗ F2(Y) + B3. [1]

Here W3 corresponds to c filters of a size n2 × f3 × f3, and B3 is a c-dimensional vector.

If the representations of the high-resolution patches are in the image domain (i.e., we can simply reshape each representation to form the patch), we expect that the filters act like an averaging filter; if the representations of the high-resolution patches are in some other domains (e.g., coefficients in terms of some bases), we expect that W3 behaves like first projecting the coefficients onto the image domain and then averaging. In either way, W3 is a set of linear filters [1].

Interestingly, although the above three operations are motivated by different intuitions, they all lead to the same form as a convolutional layer. We put all three operations together and form a convolutional neural network. In this model, all the filtering weights and biases are to be optimized [1].

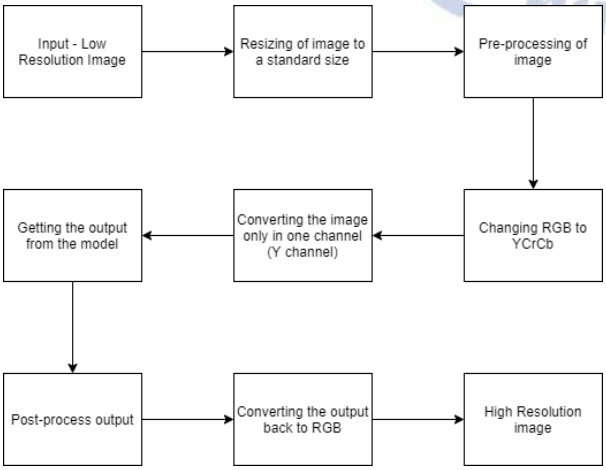


Figure 4: Block Diagram of the CNN model

EXPERIMENTS

We first study the impact of using different data sets on model performance. Next, we check the filters learned by our method. Then we explore the different architecture designs of the network and study the relationship between super-resolution performance and factors such as depth, number of filters, and filter size. Subsequently, we compared our method with the most recent state-of-the-art methods both quantitatively and qualitatively. Next, super-resolution is only applied to the luminance channel (Y channel in YCbCr color space), so c = 1 in the first/last layer, and the performance is evaluated on the Y channel (for example, PSNR and SSIM). Finally, we extend the network to process color images and evaluate the performance on different channels.

4.1 Model and Performance Trade-offs

Based on the basic network settings (i.e., f1 = 9, f2 = 1, f3 = 5, n1 = 64, and n2 = 32), we will progressively modify some of these parameters to investigate the best trade-off between performance and speed, and study the relations between performance and parameters [1].

4.1.1 Filter number

Generally speaking, if we increase the network width at the cost of running time, that is, add more filters, the performance will improve. Specifically, based on our network default settings n1 = 64 and n2 = 32, we conducted two experiments: (i) one is to use a larger network, n1 = 128 and n2 = 64, and (ii) the other It is a network that uses a smaller network n1 = 32 and n2 = 16. We also trained two models on ImageNet and tested them with an amplification factor of 3 on Set5. Obviously, superior performance can be achieved by increasing the width. However, if a faster recovery speed is required, a smaller network width is preferred, which can still achieve better performance than the method based on sparse coding (31.42 dB).

4.1.2 Filter size

In this section, we examine the sensitivity of the network to different filter sizes. In the previous experiment, we set the filter size f1 = 9, f2 = 1, and f3 = 5. The network can be expressed as 9-1-5. First, in order to be consistent with the method based on sparse coding, we fix the filter size of the second layer to f2 = 1, and expand the filter size of other layers to f1 = 11 and f3 = 7 (11-1-7). All other settings remain the same as in section 4.1. The result of a zoom factor of 3 on Set5

is 32.57 dB, which is slightly higher than the 32.52 dB reported in Section 4.1. This shows that a reasonably large filter size can grasp richer structural information and obtain better results.

4.1.3 Number of layers

Recent studies by He and Sun [16] show that CNN can benefit from a moderate increase in network depth. Here, we try a deeper structure by adding another non-linear mapping layer, which has n22 = 16 filters with a size of f22 = 1. We conducted three controlled experiments, namely 9-1-1-5, 9-3-1-5, and 9-5-1-5, respectively, in 9-1-5, 9-3-5 and 9- An additional layer was added on 5-5. The initialization scheme and learning rate of the additional layer are the same as those of the second layer. We observed that the convergence speed of the four-layer network is slower than that of the three-layer network. Nevertheless, if there is enough training time, the deeper network will eventually catch up and converge to the three-layer network.

All these experiments show that in this depth model for super-resolution, it is not "the deeper the better." It may be difficult to train. Our CNN network does not contain a pooling layer or a fully connected layer, so it is very sensitive to initialization parameters and learning rate. When we go deeper (for example, 4 or 5 layers), we find it difficult to set an appropriate learning rate to ensure convergence. Even if it converges, the network may fall into a bad local minimum, and even if there is enough training time, the diversity of the learned filters is low. This phenomenon was also observed in [7], where an improper increase in depth leads to saturation or degradation of the accuracy of image classification. Why "deeper is not better" is still an open question, which requires investigation to better understand the gradient and training dynamics in the deep architecture. Therefore, we still use a three-layer network in the following experiments.

CONCLUSION

We propose a new deep learning method for single image super-resolution (SR). We show that the traditional SR method based on sparse coding can be reformulated as a deep convolutional neural network. The proposed method SRCNN learns the end-to-end mapping between low-resolution and high-resolution images, with almost no additional pre-/post-processing other than optimization. With its lightweight structure, SRCNN has achieved superior performance than the state-of-the-art methods. We speculate that additional performance can be obtained by exploring more filters and different training strategies. In addition, the proposed structure has the advantages of simplicity and robustness, and can be applied to other low-level vision problems, such as image deblurring or synchronous SR+ denoising. People can also study a network to deal with different upgrade factors [1].

REFERENCES

1. Dong, Chao & Loy, Chen Change & He, Kaiming & Tang, Xiaoou. (2014). Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 38. 10.1109/TPAMI.2015.2439281.
2. Irani, M., Peleg, S.: Improving resolution by image registration. Graphical Models and Image Processing 53(3), 231–239 (1991)
3. Yang, C.Y., Ma, C., Yang, M.H.: Single-image super-resolution: A benchmark. In: European Conference on Computer Vision, pp. 372–386 (2014)
4. Liu, C., Shum, H.Y., Freeman, W.T.: Face hallucination: Theory and practice. International Journal of Computer Vision 75(1), 115– 134 (2007)
5. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008)
6. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
7. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: IEEE International Conference on Computer Vision. pp. 349–356 (2009)
8. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based superresolution. Computer Graphics and Applications 22(2), 56–65 (2002)
9. Timofte, R., De Smet, V., Van Gool, L.: Anchored neighborhood regression for fast example-based super-resolution. In: IEEE International Conference on Computer Vision. pp. 1920–1927 (2013)
10. Timofte, R., De Smet, V., Van Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution. In: IEEE Asian Conference on Computer Vision (2014)
11. Yang, J., Wang, Z., Lin, Z., Cohen, S., Huang, T.: Coupled dictionary training for image super-resolution. IEEE Transactions on Image Processing 21(8), 3467–3478 (2012)

12. Kim, K.I., Kwon, Y.: Single-image super-resolution using sparse regression and natural image prior. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(6), 1127–1133 (2010)
13. Dai, S., Han, M., Xu, W., Wu, Y., Gong, Y., Katsaggelos, A.K.: Softcuts: a soft edge smoothness prior for color image superresolution. IEEE Transactions on Image Processing 18(5), 969–981 (2009)
14. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with BM3D? In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2392–2399 (2012)
15. Cui, Z., Chang, H., Shan, S., Zhong, B., Chen, X.: Deep network cascade for image super-resolution. In: European Conference on Computer Vision, pp. 49–64 (2014)
16. He, K., Sun, J.: Convolutional neural networks at constrained time cost. arXiv preprint arXiv:1412.1710 (2014)