Super Resolution of the Partial Pixelated Images With Deep Convolutional Neural Network

Haiyi Mao[†], Yue Wu[‡], Jun Li[‡] , Yun Fu[‡†]
[†]College of Computer & Information Science, Northeastern University, Boston, USA
[‡]Department of Electrical & Computer Engineering, Northeastern University, Boston, USA
mao.hai@husky.neu.edu, junl.mldl@gmail.com {yuewu, yunfu}@ece.neu.edu

ABSTRACT

The problem of super resolution of partial pixelated images is considered in this paper. Partial pixelated images are more and more common in nowadays due to public safety etc. However, in some special cases, for instance criminal investigation, some images are pixelated intentionally by criminals and partial pixelate make it hard to reconstruct images even a higher resolution images. Hence, a method is proposed to handle this problem based on the deep convolutional neural network, termed depixelate super resolution CNN(DSRCNN). Given the mathematical expression pixelates, we propose a model to reconstruct the image from the pixelation and map to a higher resolution by combining the adversarial autoencoder with two depixelate layers. This model is evaluated on standard public datasets in which images are pixelated randomly and compared to the state of arts methods, shows very exciting performance.

Keywords

Super Resolution; Pixelate; Deep Learning; Adversarial Nets; Adversarial Autoencoder

1. INTRODUCTION

Super resolution, the process of from low resolution observations occupying high resolution images, has become a very tempting research topic in the past last two decades[12]. Neverthless, partial pixelates would damgage the process of super resolution. In this work, the problem is the super resolution for partial pixelated images. Figure 1 shows an example of our problem. Partial pixelated images appear more and more in people's daily life due to several reasons. The first one is public safety issues, which people want to hide personal identification information. Another reason is that image processing technologies introduce pixelates into images. Motivation of this paper is that on one hand it is necessary that in some cases we need to reconstruct the information hidden by the pixelates especially in criminal investigation. On the other hand for the super resolution issue, the performance of exsiting algorithms would be destroyed by even only small pixelates area and also makes it hard to recover information from pixlation. So regard of this, we propose a method to solve this specific problem.

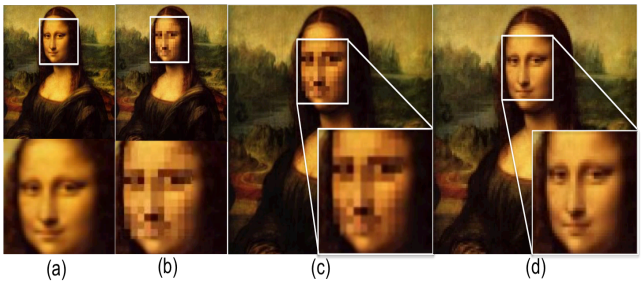

Figure 1: Illustration of the problem of partial pixelated images. (a) the image without pixelation. (b) the partial pixelated image. (c) the super reslution result of SCN [17] (scale = 2). (d) the result of DSRCNN(scale = 2). Notably our model DSRCNN can handle the partial pixelated images super resolution

Super resolution problem (SR) can be concluded as a process mapping from low resolution images observations to high resolution (HR) images observations [12]. Lately, a lot of proposed methods are aimed to solve Super Resolution. More recently, successful results have been achieved by data-driven approaches [21, 10]. However, the non-explicit form of the estimator leads to a more difficult bi-level optimization problem [10]. This difficulty can be mitigated by learning approximate inference mechanisms [9]. A generic neural network architecture is used as an inference step done by [3]. In [17], a learnt iterative shrinkage and thresholding algorithm (LISTA) network is used to approximate the sparse coding process. But none of previous model can solve partial pixelate images. In DSRCNN an autoencoder architecture is adopted, which try to learn a projection function from low resolution images to high resolutio images.

In the next section, pixelation can be formalized as a convolution operation. Naturally deconvolution can be used to solve pixelate problems. Deconvolution was studied in different fields due to its fundamentality in image restoration. For example Richardson-Lucy method [13, 18, 5, 8] is used to estimate the noise distribution for deconvolution. Another trend for image restoration is to leverage the deep neural network structure and big data to train the restoration function. For instance, [2, 19] show that a stacked denoise autoencoder (SDAE) structure [16] is a good choice

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2967235

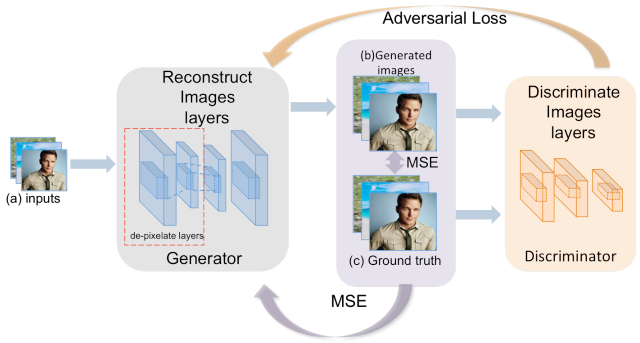

Figure 2: Architecture of DSRCNN. Input is a low resolution image with partial pixelation. Framework consists of generator and discriminator as [11]. The first two layers are depixelate layers from the idea[20].

for denoise and inpainting. Agostinelli et al. [1] generalizes it by combining multiple SDAE for handling different types of noise. In [7] and [4], the convolutional neural network (CNN) architecture [9] is used to handle strong noise such as raindrop and lens dirt. Schuler et al. [14] adds MLPs to a direct deconvolution to remove artifacts. Following [3], the deconvolution sub network are employed to do depixelate in our system. In this paper, we propose a Depixelatd Super Resolution convolutional Neural Network (DSRCNN). The DSRCNN is an end to end deep convolutional neural network framework mapping the low resolution images with partial pixelation to high resolution images. DSRCNN combines two functions depixelate with the super resolution in one adversarial autoencoder[11] architecture. The special two layers can borrow the pattern of non-pixelation area arround the pixelation to recover the original image. Figure 2 shows the general architecture of DSRCNN

2. THE ARCHITECTURE OF THE NETWORK AND ALGORITHM

The objective is to learn a mapping function from $\mathbf{Y}$ the low resolution image to an image $F(\mathbf{Y})$ and to try to make $F(\mathbf{Y})$ as similar as possible to $\mathbf{X}$, the ground truth high resolution image. The architecture is an autoencoder which is from [11] consisting of a generator and a discriminator. Generator is a network which maps from low resolution images to high resolution images. Discriminator is used to discern the ground truth and generated images. First the main loss function of whole model is proposed as following which is similar as [6]:

$$L = \min_{\Theta_1} \max_{\Theta_2} (L_G(\Theta_1) + \beta L_D(\Theta_2)), \tag{1}$$

where $L_G(\Theta_1)$ is the loss function of generator calculated by MSE. Θ_1 is the parameter sets of generator, for example, the kernel weights and biases. The same $L_D(\Theta_2)$ is the loss function of discriminator which calculated as cross-entropy named as adversarial loss. Θ_2 is the parameter sets of discriminator, for example, the kernel weights and biases. Finally, the loss function is a linear combination of L_G amd L_D, where β is setting to 0.01 empirically. In general, the model is trying to minimize MSE in generator and maximize the discriminator loss. The following is the details of every term in the loss function.

2.1 Generator(Super Resolution)

Super resolution is modeled as a process mapping from low resolution to high resolution [3]. The loss function of this part is modeled as follows:

$$L_G(\Theta_1) = \sum_{i=1}^{n} ||F(\mathbf{Y}_i; \Theta_1) - \mathbf{X}_i||^2, \tag{2}$$

Where Θ_1 is the set of parameters of the model, n is the number of samples.

2.2 Depixelate Network

Pixelate operation can be formalize as a linear convolution as shown in Eq.(3). In the light of this, we adopt previous work [20] which proposes a special convolutional network that approximates a deconvolution operator.

$$\mathbf{Y} = \mathbf{Y}' * \mathbf{K} \tag{3}$$

, where $\mathbf{Y}'$ is the image with pixelate, $\mathbf{Y}$ is the high resolution image without pixelates. $\mathbf{K}$ refers to the convolutional kernel function which is a average function. From previous work the kernel separability is achieved via singular value decomposition (SVD)[22]. Given the inverse kernel $\mathbf{K}^\dagger$, decomposition

$$\mathbf{K}^\dagger * \mathbf{Y}' = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{4}$$

Where $\mathbf{K}^\dagger$ is the deconvolutional kernel, and $\mathbf{Y}'$ is the image with pixelates. We denote by u_j and v_j the j^{th} columns of $\mathbf{U}$ and $\mathbf{V}$, s_j the j^{th} singular value. So we have

$$\mathbf{Y}_i = \sum_j s_j \cdot u_j * (v_j^T * \mathbf{Y}'_i), \tag{5}$$

where $\mathbf{Y}_i$ is the i^{th} image, $\mathbf{Y}'_i$ is the i^{th} image with pixelates.

In conclusion, depixelate problem can be written in following:

$$F(\mathbf{Y}_i; \Theta_1) = F(\sum_j s_j \cdot u_j * (v_j^T * \mathbf{Y}'_i), \Theta_1), \tag{6}$$

by combining Eq.(1) and Eq.(3), we have

$$\begin{aligned} L_G(\Theta_1) &= \sum_{i=1}^{n} ||F(\mathbf{Y}'_i, \Theta_1) - \mathbf{X}_i||^2 \\ &= \sum_{i=1}^{n} ||F(\sum_j s_j \cdot u_j * (v_j^T * \mathbf{Y}'_i), \Theta_1) - \mathbf{X}_i||^2 \end{aligned} \tag{7}$$

According to [20] we can use 1×121 and 121×1 special kernels to approximate the SVD decomposition.

2.3 Adversarial Loss

However if we only use the super resolution mapping network and the deconvolutional operators, the generated images have gaussian noise brought by the objective function and sometimes it is hard to get converged. In the light of generative advesarial nets [6] and [11] , we add one more discriminator in the architecture. Naturally the loss function is a combination of two loss as shown in the former equation

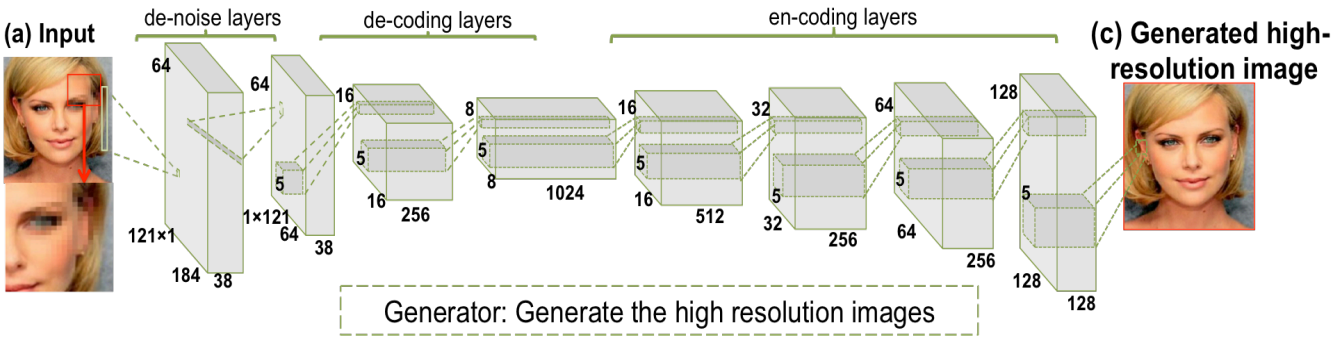

Figure 3: Details of the generator.Generator consists of de-noise layers, encoding layers, and decoding layers.

shown in Eq.(1). So we have two loss functions:

$$L_D(\Theta_2) = -\frac{1}{n}\sum_{i=1}^{m}[\mathbf{X}_i \ln D(F(\mathbf{Y}_i';\Theta_1);\Theta_2) + (1-\mathbf{X}_i)\ln(1-D(F(\mathbf{Y}_i';\Theta_1);\Theta_2)] \quad (8)$$

where Θ_2 is the set of parameters in discriminator. L_D is the loss function of discriminator, which is cross-entropy.

Eq.(2) is the MSE calculated by generator, and Eq.(8) is the loss function of the discriminator. Eq.(1) from [6] is the main loss function of the image generator, which is a linear combination of Eq.(4) and Eq.(5).

The discriminator's architecture is the same as [11]. Kernels size are, $128\times5\times5$, $256\times5\times5$, $512\times5\times5$, $1024\times5\times5$, with padding 2 except first two layers. The loss of discriminator is cross-entropy which is given in Eq.[5] .

So the whole framework is one to minimize the MSE of generator loss then maximize the discriminator loss. During the training phases, the depixelate layers recovers the original information then the generator constructs the super resolution image. Then generated image and ground truth are both inputs of the discriminator. The discriminator will generate two scores for the ground truth with generated image and the ground truth with itself. The cross entropy of the two scores is used as the loss of the generator. However, during the testing phase, higher resolution images are only created by generator, no more discriminator gets involved.

3. EXPERIMENTS

We evaluate our models on two public available standard data sets for super resolution. In this section, we first give a detailed decription of these data sets and how we add pixelation on the images. Then, the evaluation protocol used in the experiments is introduced. Next, the performance of several models is reported and some images as well.

We trained three scale sizes of network, they are 2, 3, 4. For the fair comparison the standard datasets Set5 and Set14 are used to evaluate the performance for our model and other methods.

3.1 Data Sets

The same training sets, test sets, and protocols are adopted as in [15] when compared with traditional example-based methods and CNN based methods. Be specific, the training set includes 1091 images which are from [3]. In addition for the 1091 images training set, we add pixelation square area randomly on all the training images with the size of the 0.2 times size of images. For every single image the pixelate operation would be done 4 times which means one image has 4 pixelated versions with different pixelation area location as inputs in training datasets. In order to evaluate the performance, upscaling factors 2, 3 are imposed on Set5 (5 images) and upscaling factor 2 and 3 are imposed on Set14 (14 images) to investigate the results. We compare DSRCNN with the Super Resolution methods related to convolutional neural network: SRCNN[3] and SCN[17]. Moreover K-SVD[22], and bicubic interpolation are also included in the comparison. The implementations are all from the publicly available codes provided by the authors.

Table 1: The result of Set5 measured by PSNR value. The comparison methods are Bicubic interpolation, SCN[17], SRCNN [3], K-SVD [23].

Set5 images (scale = 2)	Bicubic	SCN	SRCNN	K-SVD	DSRCNN
baby	33.91	34.79	35.03	34.59	**35.11**
bird	32.58	34.70	34.50	34.23	**35.06**
butterfly	24.04	27.31	27.30	26.65	**27.38**
head	32.88	33.55	33.41	33.23	**33.82**
woman	28.56	30.79	30.54	30.46	**30.79**
Set5 images (scale = 3)	Bicubic	SCN	SRCNN	K-SVD	DSRCNN
baby	32.49	32.95	33.85	32.89	**33.90**
bird	30.48	30.99	32.20	32.17	**32.51**
butterfly	23.80	24.01	24.08	24.89	**24.59**
head	32.24	32.39	32.40	32.33	**32.54**
woman	26.96	29.97	29.82	29.83	**30.41**
Set5 images (scale = 4)	Bicubic	SCN	SRCNN	K-SVD	DSRCNN
baby	31.78	33.09	33.13	32.25	**33.54**
bird	30.18	32.18	32.52	31.65	**32.37**
butterfly	22.10	25.06	25.46	24.76	**25.63**
head	31.59	32.55	32.44	31.62	**32.92**
woman	26.46	28.89	28.59	28.31	**28.91**

3.2 Training Details

Specifically, in the training phase, the ground truth images are prepared as 184 pixel 6 subimages randomly cropped from the training images. Every image is pixelated with $0.2\times width, 0.2\times height$ rectangular area. And the location of the rectangular is randomly picked. Moreover, for ev-

Figure 4: Results of woman. The comparison methods are Bicubic interpolation, SCN [17], ground truth and DSRCNN

Table 2: The result of Set14 measured by PSNR value. The comparison methods are Bicubic interpolation, SCN[17], SRCNN [3], K-SVD [23].

Set14 images (scale=2)	Bicubic	SRCNN	SCN	K-SVD	DSRCNN
banboo	24.13	24.86	24.64	24.26	**25.49**
babara	27.66	28.00	27.84	27.72	**28.29**
brindge	23.39	**26.58**	23.54	24.04	25.15
costguard	28.06	29.12	28.81	28.91	**30.58**
comic	24.62	**26.02**	25.63	25.62	28.65
face	34.06	34.83	34.42	34.33	**35.78**
flowers	28.82	30.37	29.67	28.91	**33.33**
foreman	32.05	34.14	33.26	33.12	**34.89**
lenna	32.65	34.70	33.25	33.13	**36.57**
man	28.13	29.25	29.00	28.51	**30.95**
monarc	29.97	32.94	30.47	29.76	**37.63**
peper	32.29	**34.97**	33.48	32.88	34.83
ppt3	25.85	26.87	27.45	26.13	**30.52**
zebra	27.77	30.63	28.15	27.74	**33.45**

Set14 images (scale=3)	Bicubic	SRCNN	SCN	K-SVD	DSRCNN
banboo	23.05	23.44	23.34	23.26	**23.58**
babara	26.20	26.46	25.96	25.18	**26.65**
brindge	23.82	24.47	24.63	24.04	**25.37**
costguard	25.84	26.36	26.22	25.91	**27.18**
comic	22.92	**24.18**	23.75	23.62	23.95
face	32.53	33.31	33.30	32.91	**33.47**
flowers	26.79	28.44	**28.48**	28.14	28.40
foreman	30.68	32.65	31.64	31.12	**33.32**
lenna	31.25	32.65	32.18	31.73	**33.12**
man	26.73	27.94	28.21	27.51	**28.33**
monarc	27.53	28.81	30.67	28.76	**28.94**
peper	32.00	**33.86**	32.43	31.88	33.83
ppt3	23.14	25.71	25.74	25.13	**25.69**
zebra	26.19	28.33	28.15	27.97	**28.54**

ery single image the pixelate operation will be done 4 times which means one image has 4 pixelated images as inputs in training datasets. In total, we have around 4k images to train the model. In order to get more solid model, we pretrained the discriminator and generator with the code and data given by [11]

We extract subimages from original images by a stride of 14 the same as[3]. So the input images are 184×184 in RGB 3 channels. The kernel size in generator layers are 38 kernels with size of 121×1, 38 kernels with size of 1×121, other kernel size are shown in the figures 3.

3.3 Results

The PSNR values are evaluated in the testing data sets Set5 and Set14 shown in the table 1 and table 2. For the Set5 is tested under scale 2, 3 and 4, Set14 is tested under scale 2 and 3. Additionaly, the image results of woman and butterfly in Set5 are also shown in the next page. From the results, we can see the DSRCNN out-performs all images in Set5. For Set14 few images, for example comic, peper, flowers, fail to out perform SRCNN or SCN. The reason of failure is because images are relatively large to our kernel size, which leads that less information is gathered in deconvolutional phases. Generally, from the results, DSRCNN out-performs the previous algorithms. To extend, more work can be done which is based on different pixelation size, pixelation area sizes, or other kinds of blurriness.

Acknowledgement

This research is supported in part by the NSF CNS award 1314484, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

4. REFERENCES

[1] F. Agostinelli, M. R. Anderson, and H. Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *Advances in Neural Information Processing Systems*, pages 1493–1501, 2013.

[2] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2392–2399. IEEE, 2012.

[3] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014*, pages 184–199. Springer, 2014.

[4] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 633–640, 2013.

[5] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (TOG)*, 25(3):787–794, 2006.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[7] V. Jain and S. Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009.

[8] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041, 2009.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[10] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):791–804, 2012.

[11] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[12] K. Nasrollahi and T. B. Moeslund. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6):1423–1468, 2014.

[13] W. H. Richardson. Bayesian-based iterative method of image restoration*. *JOSA*, 62(1):55–59, 1972.

[14] C. Schuler, H. Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1074, 2013.

[15] R. Timofte, V. Smet, and L. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1920–1927, 2013.

[16] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[17] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 370–378, 2015.

[18] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*, volume 2. MIT press Cambridge, MA, 1949.

[19] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012.

[20] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014.

[21] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[22] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. Springer, 2010.

[23] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 479–486. IEEE, 2011.