International Journal of Computer Applications (0975 – 8887)
Volume 173 – No.3, September 2017

Super-Resolution using Sub-pixel Recursive Adversarial Network with Perceptual Loss

Ganesh Singh Bisht
Faculty of Technology
Uttarakhand Technical University
Dehradun, Uttarakhand

Pawan Kumar Mishra
Faculty of Technology
Uttarakhand Technical University
Dehradun, Uttarakhand

ABSTRACT
Super-Resolution is a classic problem in computer vision and many methods have been designed to reconstruct the high-resolution image from low-resolution image. Recent solution of such problem is based on the convolutional neural network where mapping function is used to map low-resolution image to high-resolution image based on per-pixel loss or mean-square error. We introduce a framework that uses perceptual loss function and provides much finer results with high improvement in speed. This framework also replaces the bicubic interpolation for upscaling image with the sub-pixel convolutional layer that learns upscaling filters to upscale the low-resolution feature map to the high-resolution image, that leads less computational complexity. The Proposed method also deals with high upscale factor, by the introduction of an adversarial network that helps in recovering finer texture details in a low-resolution image.

General Terms
Computer Vision, Super-Resolution

Keywords
Super-resolution, deep learning, convolutional neural network, perceptual loss, adversarial network, sub-pixel convolutional layer.

1. INTRODUCTION
Super-resolution is a phenomenon of retrieving high-resolution image from one or more low-resolution images, has been trending research topic over the years. The aim is to get high pixel density with more details information about original scene rather them sampling imaging device. It has many application areas that solve many real-world problems, from satellite imaging to aerial imaging, medical imaging to face recognition, surveillance to number plate reading, text analysis, and biometric recognition, and much more. Many research papers have been written aiming to get a new super-resolution algorithm for a specific purpose [1], [2], [3].

The common approach to solve super-resolution problem is considering low-resolution image is created by resampling high-resolution image. A low-resolution image is created by series of degradation process such as warping, blurring, decimation and added noises. The image observation model is necessary for the recovery of the high-resolution image as incorrect model degrade image further.

Super-resolution methods are categories in three ways – the first one is interpolation based methods, second is reconstruction based methods and the last one is example based methods. Interpolation is the easiest one where missing pixel values are interpolating into an image by neighboring pixel information. Example based methods are well known state-of-art method where external dictionary of correspondence high and low-resolution patch images is used to find a high-resolution image by learning mapping from patches.

Convolutional Network is trained by calculating and evaluation of minimum loss function [4], [6]. Perceptual Loss function generates a high quality image by taking a difference of high-level image feature rather than a difference between pixel [5]. Proposed approach to solving the problem of super-resolution is by training feed-forward convolutional neural network. Dong et al used this approach using per-pixel loss function to measure the difference between high-resolution image and ground truth image. Their approach doesn't cover the perceptual difference between image. For two identical images with one pixel offset in first, they vary differently as measured by per-pixel despite their perceptual similarity. In this work, the approach of feed forward neural network combines with perceptual loss function to train the network by measuring image similarities rather than pixel-loss.

In CNN bi-cubic interpolation takes place to upscale the image. It is a pre-processing operation for most of the super-resolution task. Proposed method doesn't use bicubic interpolation as it increases computational complexity, this framework extracts the feature map in low-resolution space and add a sub-pixel convolutional layer that learns upscaling filters to upscale feature maps into high-resolution output.

For high upscale factors, CNN produces recover images with lost finer texture details. To solve the problem, the adversarial neural network is used that uses the deep network with skip connection and uses perceptual loss with discriminator that makes output images hard to distinguish from an original high-resolution image.

2. RELATED WORK
Super-Resolution is classified as single-image and multi-image super-resolution based on a number of input images. In the multi-image super resolution [7], [8], multiple images of same scene having different pixel alignments and shifted sub-pixel containing different information about same scenery is used to get complementary information between them by non-linear interpolation and deblurring. Multiple images are not always available for super-resolution that leads to a limited number of low-resolution information which leads to single image SR.

Our focus here is single image super resolution. The first method that was introduced to tackle single image super-resolution is prediction base i.e. linear and bicubic method [9], which is fast but doesn't result in finer, smooth and good quality texture, that leads to the introduction of Edge preservation based method [10], [11]. Most of the modern approach is based on example pair or patches that learn mapping from the low-resolution image with the high-resolution image and rely on training data.

Earlier work presented by Freeman Et al [12], [13]. Self-similarity is introduced by Huang et al [14], where prior dictionary was used. Gu et al [15], introduced the sparse coding approach that improved super-resolution by not relying on patches rather than processing the complete input image. Neighbor embedding technique where introduced that find similarity between low resolution training patches and combine to find high-resolution image [16], [17]. Regression method to solve regression problem was introduced such as regression [18], tree [19] and random forest [20].

Recently, convolutional neural network based super resolution method provide state of art performance. Dong et al [21], [22], gives the first deep convolutional neural network that used 3 convolution layer with bicubic interpolation to learn end to end mapping between LR and HR image. Sparse representation based SR method introduced by Wang et al [23]. To increase the performance deep-recursive convolutional neural network presented by Kim et al [24].

Many recent papers have used perceptual optimization for to extract high levels of feature extraction to maximize class prediction scores [25] or individual features [26] in trained networks. Yang et al [27], group super resolution techniques into prediction-based methods, edge based methods, statistical methods [28], [39], [30], patch based methods and sparse dictionary methods.

3. MOTIVATION

Paste decades reveals many super-resolution techniques that have computation time vs resolution trade off. Some of them result in better resolution with a large amount of computation time and other with less computation time by slight compromise in resolution. Lately, Deep learning emerged as savior and created a big impact on many areas such as machine learning, natural language processing, artificial intelligence, computer vision and signal processing. Deep learning is revolving around the neural network and here a good method is presented that uses the deep neural network to reconstruct a high-resolution image.

4. METHOD

The aim of super-resolution is to reconstruct high-resolution image Y given a low-resolution image X, generated by downscaled from corresponding original ground truth image G. To synthesize low-resolution image X, a Gaussian filter is used then downsample the resultant image by a factor R known as upscaling ratio. Each of low and high-resolution image has color channel C thus, ground truth image can be represented as H x W x C and low-resolution image as HR x WR x C.

SRCNN proposed a solution to solve super-resolution by recovering image from bicubic upscale and interpolated image B instead of a low-resolution image with help of 3-layer convolutional neural network. Proposed approach is different in a sense that we avoid upscaling and interpolating low-resolution image before entering it into a convolutional neural network. Here, a convolutional operation is performed on the first layer to a low-resolution input image, and then a sub-pixel convolution operation on next layer that will upscale the feature maps created from low-resolution image to produce high-resolution image Y. Bicubic interpolation causes complex computation overhead, removing it completely from neural network pipeline result in better accuracy with little computation time.

A network that consists of L layer, first layer L_1 can be express as follows:

$$L_1(X, W_1, B_1) = \emptyset\, W_1 * X + B_1,$$

$$L_{n-1}(X, W_{n-1}, B_{n-1}) = \emptyset\, W_{n-1} * L_{n-1}\, X + B_{n-2}$$

Here W_l, B_l are weight and biases respectively. W_l is a weight of size $f_1 \times f_l \times k_1 \times k_1$, where f_1 is the number of features at layer l and k_1 is the filter size at layer L_l. Activation function φ is applied element-wise and is fixed. Last layer L^n converts feature maps to high resolution image X.

4.1 Deconvolution Layer

In convolution layer, multiple activation filters are combined together to form a single activation. In contrast to this in deconvolution layer, a single input activation results in multiple activation filters. This is fully convolution and deconvolution layer as no pooling is done in the framework. The pooling can eliminate the image details that are required for image restoration process. Using deconvolution layer is proved better than applying only fully convolutional neural network. In fully convolution layer noises are remove step by step in each of convolution layer that results in loss of frequency information. So in neural network convolution maintain the image content and deconvolution recover the details information. This technique is very popular to generate high-level features and semantic information from an input image. Bicubic interpolation is a special kind of deconvolution layer that proposed in SRCNN. This can be simulated as element-wise multiplication of filter by each input pixel with some stride s.

4.2 Sub-Pixel Convolution Layer

One way to scale a low-resolution image by a factor f is by applying convolution operation on input low-resolution image with a stride of 1/f. The resultant image is scaled imaged in high-resolution space followed by series of convolution, pooling, activation function with a stride of 1 to form a high-resolution image. Initial convolution operation is implemented by interpolation and unpooling. This result in high computational cost by a factor of f^2 because of convolution is done in high-resolution space.

Another way of upscaling image is by applying filter W of size k with a stride of 1/f in a convolution operation. The filter that falls between pixels are not needed to be calculated and just need to be avoided. All other parts of filter will use in convolution operation. The total calculated activation is f^2 which is equivalent to bicubic interpolation. The activation patterns depend on different sub-pixel location during convolution operation by applying the filter to an input image and can be expressed as modulus of output pixel with scaling factor f i.e. mod (x, f) and mod (y, f). This is implemented with a modulus of kernel or weight with scale factor and can be expressed as:

$$Y = L(X) = S\, W_N * L_{n-1}(X) + B_N$$

where S is a periodic shuffling operator that upscale elements of a H × W × C to f H × f W × C with scale factor f. W_N is convolution operator. This operation doesn't exist for the last layer. It is easy to see that when kX = kY f and mod (kx, f) = 0 is sub-pixel convolution in the LR space with the filter W. this layer acts as the sub-pixel convolutional layer and produce the high-resolution image from low resolution features maps by applying one upscale filter with each feature maps.

It is noticeable that the implementation of the above periodic shuffling can be avoided in training time. Instead of shuffling the output as part of the layer, one can reshuffle the training data to match the output of the layer before PS.

4.3 Super-Resolution with Perceptual Loss

Many recent papers show the use of perceptual optimization to generate an image with high-level features extracted from a convolutional neural network based on class prediction scores or individual features. For a scale factor of 4 or 8, a low-resolution image doesn't contain any fine details to generate a high-resolution image. Per-Pixel loss function can't be used for such a large scale factor. To solve this problem, the per-pixel loss is not used to train the convolution network instead of this semantic knowledge of input image is carried forward in neural network as large-scale factors need more sematic knowledge about the input.

The most popular metrics to evaluate the performance of super-resolution method are PSNR and SSIM, which are largely depends on pixels' values and doesn't correlate with human visual perception. PSNR and SSIM operate by making an assumption of Gaussian noise and depends on the difference between pixels of images that may be insufficient for measuring the performance of super-resolution. PSNR value is always high for a neural network train to minimize per-pixel loss rather than training to minimize feature loss.

So the main aim of proposed network is to minimize the feature difference between high-resolution image with ground truth image rather than just to achieve state of art PSNR and SSIM values.

4.4 Perceptual Loss Functions

Loss functions give feedback to the neural network whether the neural achieve its expected output or not. This help neural network to reinforce the concept and result in better training. Loss functions can be defined in such a way that it can measure the high level perceptual difference as well as the semantic difference between images. This is perceptual loss function for the network and acting themselves as a deep convolutional neural network.

Instead of matching pixels of output image Y' to ground truth image, it is better to compute feature representations by the neural network φ. For the i^{th} layer of the network, φi(x) be the activations and it is a convolutional layer for input image X. φi(x) also features map for i convolutional layer with dimension H_i x W_i x C_i. The feature reconstruction loss can be expressed in term of Euclidean difference between feature representation:

$$\boldsymbol{\ell}^{\emptyset}(\boldsymbol{y}', \boldsymbol{y}) = \frac{\mathbf{1}}{\boldsymbol{HWJ}} ||\emptyset(\boldsymbol{y}') - \emptyset(\boldsymbol{y})||\ {}^{\mathbf{2}}_{\mathbf{2}}$$

Network aiming to minimize feature reconstruction loss will result in production of reconstructed image y' that are similar to y in term of feature and semantic values. Early layers will be benefitted by this loss function and as we increase layer in neural network color, shape and texture are not preserved. Use of feature reconstruction loss in image transformation network always synthesizes output image y' that perceptually similar to ground truth image y.

4.5 Simple Loss Functions

Two additional loss functions are also introduced that depend on pixel information.

Pixel Loss: It is pixel by pixel Euclidean distance between pixel the output image y' and ground truth image y. For an image with dimension H x W x C, pixel loss can be expressed as:

$$PL(y', y) = \frac{\|y' - y\|_2^2}{WHC}$$

This expression can be used if we have ground-truth image y to be matched by the network.

Total Variation Regularization. For spatial smoothness in the output image y', the network makes use of total variation regularizer to encourage spatial smoothness in the output image ŷ, we follow prior work on feature inversion and super-resolution and make use of total variation regularizer l_{TV} (y').

4.6 Adversarial Network

It is a powerful network for generating a natural image with high perceptual quality. Generative adversarial network optimized for the perceptual loss that uses ResNet architecture. Use of a discriminator network D θ D which optimize in an alternating manner along with G θ G [31]. The general idea behind this formulation is that it allows one to train a generative model G with the goal of fooling a differentiable discriminator D that is trained to distinguish super-resolved images from real images. With this approach, the generator can learn to create solutions that are highly similar to real images and thus difficult to classify by D.

This encourages perceptually superior solutions residing in the subspace, the manifold, of natural images. This is in contrast to SR solutions obtained by minimizing pixel-wise error measurements, such as the MSE.

4.6.1 Adversarial Loss

In addition to the perceptual losses described so far, method also adds Adversarial loss to GAN network. This is a favourable solution for network by trying to SR fool the discriminator network. The generative loss is defined based on the probabilities of the discriminator D_D (G_G (X)) over all training samples as:

$$AL = \sum_{n=1}^{N} - log\ D_D\ (G_G\ X))$$

Where, D_D (G_G (X)) is the probability that the reconstructed natural HR image.

4.7 Image Transformation Networks

This image transformation networks do not use pooling layer, the stride is used for downsampling and upsampling of an image in convolution network. This network consists of convolutional layers followed by Rectified Linear Unit and batch normalization and output layer. Output layer is scaled by tanh to make sure that its pixels in the rage of 0-255. All the convolutional layers have filter of size 3 x 3 except first and last layers that use kernel of size 9 x 9.

Inputs and Outputs: For an input image of W x H x C with upscaling factor f, the output is a high-resolution image of dimension f W × f H × c. The image transformation neural network is fully-convolutional so one can apply this method to images of any resolution.

Downsampling and Upsampling: For super-resolution with an upsampling factor f, to upscale the low-resolution image by use of several residual blocks followed by $\log_2$ f convolutional layers with stride ½, rather than use bicubic interpolation to up-sample the low-resolution input before input into the network.

5. EXPERIMENT

The implementation of the framework is done in python with keras library. All the test images are taken from set 5 data sets. Many test images are being used to show the benefit of approach used in this paper. All the function for super-resolution is written in python 3.X using Keras - neural network and deep learning framework with additional libraries such as numpy, scikit-learn, scipy and matplotlib. Many inbuilt functions of Keras are used for development of different stages of implementation. A test image which is the low-resolution image is input to the system and high-resolution image with more detail will return as output with the scale factor.

5.1 Training Data

The training dataset consist of 91 images and size of each training images is 33 x 33. 91 images are decomposed in 24,800 images. Set-5 as well as large dataset Set-15 is used as test image. The super-resolution performance further increases by using large training set but computation cost will be high. It has also been observed that 91 images are sufficient to give variability in images.

5.2 Training Image

Total 91 images data set are used to create training images. Each of training images is a sub-image formed after cropping randomly the high-resolution image. The sub image is f_{sub} X f_{sub} in size and are not overlapping images. To produce low-resolution image these sub-images are further gone into blurring process by Gaussian kernel of sigma 0.5. There is no padding in image to avoid border effects during training. These sub-images then downscaled to 1/3rd of size and upscaling again to form 33 x 33 size of sub-images.

5.3 Filtered Learned

Each of layer has some sort of filtered learned by the neural network. Filtered learned in the first layer is Gaussian, Laplacian, edge detectors, texture detectors and others. Feature map of the first layer contains different structures and each of further layer contains features map with more intensities of these filters.

5.4 Training

There are three types of layer in this framework convolution, deconvolution and merge or sum layer followed by Rectified Linear Unit. Let us have an image X, convolution layer can be expressed as:

$$F(X) = \max(0, W_i * X + B_i)$$

Here Wi is weight, Bi is biases and * is convolution operation.

For the sum or merge layer the output is sum of two input images of same size followed by Rectified Linear Unit activation function. This can be expressed as:

$$F(X_1, X_2) = \max(0, X_1 + X_2)$$

The estimation of weight or kernel learned by an end to end mapping is kernel of convolution. For an input image X and out image Y the mean square error is summarized as:

$$\mathrm{L}(\emptyset) = \frac{1}{N}\sum_{i=0}^{n} ||F(X^i;\, \emptyset) - Y^i||_F^2$$

This framework learns mapping function from low resolution to high-resolution image by applying the bridge connection between input and output images rather than simply learns from corrupted image to high-resolution image. We found that optimizing the network for low-resolution image is converge better than the high-resolution image. Network is implemented and train using keras with theano support and Adam optimizer is used for better learning rate and convergence rate. The learning rate for all the layers are basically same.

5.5 Testing

Proposed network train on local sub-images and it can perform image super-resolution on any arbitrary size of an image. Taking a testing image in model outperforms all existing methods. To apply the deep learning on low computational devices the testing speed should speed up. Down-sampling the feature map will reduce the size and up-sampling it in the corresponding layer will gain back the original size of the feature map. This leads to the good testing efficiency with negligible degradation in performance.

This Framework uses a stride of 2 for down-sampling feature map and test image on i5 1.6 GHz processor. The PSNR is slight degrade by down-sampling feature map. For the first convolution layer, the feature map size is reduced by ¼ that speed up testing process with slight 0.1 degrades in PSNR value.

6. RESULTS

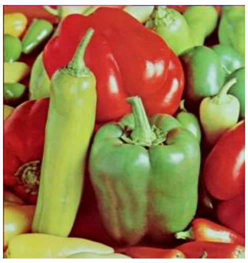
Following are the five test images being used for testing proposed method, taken from image set 4 and set 14.

(a)

(b)

(c)

(d)



(e)

Fig 1: Test images from set 4 and set 14. a) baby b) butterfly, c) head, d) woman, e) vegetable image

Bicubic SRCNN Proposed

Fig 2: Result of super-resolution on test image 1 (baby) using bicubic, convolutional neural network (SRCNN) and proposed method with upscaling factor 2

Bicubic SRCNN Proposed

Fig 3: Result of super-resolution on test image 2 (butterfly) using bicubic, convolutional neural network (SRCNN) and proposed method with upscaling factor 2

Bicubic SRCNN Proposed

Fig 4: Result of super-resolution on test image 3 (head) using bicubic, convolutional neural network (SRCNN) and proposed method with upscaling factor 2

Bicubic SRCNN Proposed

Fig 5: Result of super-resolution on test image 4 (woman) using bicubic, convolutional neural network (SRCNN) and proposed method with upscaling factor 2

Bicubic SRCNN Proposed

Fig 6: Result of super-resolution on test image 5 (vegetable) using bicubic, convolutional neural network (SRCNN) and proposed method with upscaling factor 2

Table 1. Comparison Table for Parameter

Parameter	Image	Bicubic	SRCNN	Proposed
PSNR	1	29.1834	30.3430	30.3543
	2	23.8088	25.0765	25.5323
	3	34.2084	35.4408	35.5042
	4	26.3540	27.7330	28.0234
	5	31.5643	33.1242	34.8704
MSE	1	78.4767	60.0875	58.2341
	2	270.519	202.036	180.345
	3	24.6742	18.5781	18.2012
	4	150.549	109.592	101.2344
	5	45.5614	40.9835	37.4534
SSIM	1	0.8552	0.8855	0.8870
	2	0.8662	0.8977	0.9021
	3	0.8750	0.8966	0.9023
	4	0.8870	0.9146	0.9175
	5	0.8641	0.8836	0.8956
NCC	1	0.9975	0.9968	0.9968
	2	0.9785	0.9822	0.9825
	3	0.9973	0.9960	0.9958
	4	0.9888	0.9903	0.9900
	5	0.9747	0.9862	0.9913
AD	1	-0.4828	-0.1508	-0.1323
	2	-0.4042	0.0496	-0.0399
	3	-0.2135	0.1169	0.0056
	4	-0.2282	0.0635	0.1146
	5	-0.2674	0.0345	0.1267
SC	1	1.0019	1.0041	1.0043
	2	1.0300	1.0259	1.0250
	3	1.0027	1.0061	1.0065
	4	1.0152	1.0142	1.0150
	5	1.0237	1.0241	1.0245
MD	1	62	62	62
	2	121	107	100
	3	45	37	36
	4	81	83	80
	5	78	75	74
NAE	1	0.0381	0.0336	0.0329
	2	0.0742	0.0635	0.0501
	3	0.0478	0.0433	0.0426
	4	0.0531	0.0453	0.0440
	5	0.0347	0.0328	0.0319

A number of metrics were used to show the benefit of proposed method in this paper. A) PSNR – Peak signal to noise ratio, B) SSIM – Structural Similarity Index, C) NCC - Normalized Cross-Correlation, D) AD - Average Difference, E) SC - Structural Content, F) MD - Maximum Difference, G) NAE - Normalized Absolute Error.

7. CONCLUSION

In this work, we have presented a super-resolution method using a deeply-recursive convolutional network which efficiently reuses weight parameters while exploiting a large image context. Here, the feature extraction stage is performed in the LR space instead of HR space that reduce the computational complexity of overall operation. To bypass bicubic interpolation a sub-pixel convolution layer is used that can compute super-resolution in HR space rather than LR space, with little computation. Perceptual loss function is used instead of per-pixel loss that give benefits of both neural network transformation networks and optimization based methods. This framework is capable of inferring natural images for 4x upscaling factors. We generate high quality images by defining and optimizing perceptual loss functions based on features extraction.

In future scope, one can explore this method with other computer vision problem and image transformation tasks such as colorization, classification, face recognition, object detection and image segmentation. One can try more network architecture to get best and optimized network. Additional performance can be increase by adding more number of filters and better network structure. One can try more recursions in order to use image-level context. We believe our approach is readily applicable to other image restoration problems such as denoising and compression artifact removal.

8. REFERENCES

[1] D. Glasner, S. Bagon, and M. Irani. 2009. Super-resolution from a single image. International Conference on Computer Vision (ICCV).

[2] Xiong, Z., Sun, X., Wu, F. 2010. Robust web image/video super-resolution. Image Processing, IEEE Transactions on 19(8) 2017–2028.

[3] J. Yang, J. Wright, T. Huang, and Y. Ma. 2010. Image super-resolution via sparse representation. IEEE Transactions on Image Processing, pages 2861–2873.

[4] J. Kim, J. K. Lee, K. M. Lee. 2016. Deeply recursive convolutional network for image super-resolution. Computer Vision and Pattern Recognition.

[5] J. Johnson, A. Alahi, L.F. Fe. 2016. Perceptual Losses for real time style transfer and super-resolution. Computer Vision ECCV.

[6] M. Liang and X. Hu. 2015. Recurrent convolutional neural network for object recognition. CVPR.

[7] S. Borman and R.L. Stevenson. 1998. Super resolution from image sequence – a review. Midwest Symposium on circuits and systems, pages 374-378.

[8] S. Fasiu, M. D. Robinson, M. Elad, and P. Milanfar. 2004. Fast and robust multiframe super resolution. IEEE transaction on image processing. Pages 1327-1344.

[9] Irani, M. Peleg, S. 1991. Improving super-resolution by image registration. Graphical models and image processing, pages 231-239.

[10] Freedman, G. Fattal, R. 2011. Image and video upscaling from local self-examples. ACM Transaction on graphics.

[11] Sun J., Xu Z, Shum H. Y. 2008. Image super-resolution using gradient profile prior. Computer Vision and Pattern Recognition, IEEE.

[12] W.T. Freeman, T.R. Jones, and E.C. Pasztor, O.T. Carmichael. 2000. Learning low level vision. International Journal of Computer Vision, pages 25-27.

[13] W.T. Freeman, T.R. Jones, E.C. Pasztor. 2002. Example based super-resolution. IEEE Computer Graphics and Applications, pages 56-65.

[14] J. B. Huang, A. Singh, and N. Ahuja. 2015. Single image super-resolution from transformed self-exemplars. IEEE Conference on Computer Vision and Pattern Recogniton, pages 5197-5206.

[15] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng and L. Zhang. 2015. Convolutional sparse-coding for image super-resolution. IEEE, International conference on computer vision, pages 1823 – 1831.

[16] R. Timofte, V.De Smet, and L. Van Gool. 2013. Anchored neighborhood regression for fast example-based super- resolution. IEEE, International conference on computer vision, pages 1920-1927.

[17] R. Timofte, V. De Smet and L. Van Gool. 2014. A+: Adjusted anchored neighborhood regression for fast super-resolution. Springer, Asian Conference on computer vision, pages 111-126.

[18] H. He and W.C. Siu. 2011. Single image super-resolution using Gaussian process regression. IEEE, conference on computer vision and image processing, pages 449-456.

[19] J. Salvador and E. Perez – Pellitero. 2015. Naive bayes super-resolution forest. International conference on computer vision, pages 325-333.

[20] S. Schulter, C. Leister, and H. Bischof. 2015. Fast and accurate image upscaling with super-resolution forest. IEEE, Conference on computer vision and pattern recognition, pages 3791-3799.

[21] C. Dong, C.C. Loy, K. He and X. Tang. Learning a deep convolutional network for super resolution. Springer, European conference on computer vision. pages 184-199.

[22] C. Dong, C.C. Loy, K. He and X. Tang. 2016. Image super-resolution using deep convolutional networks. IEEE Transactions on pattern analysis and machine learnings, pages 295-307.

[23] Z. Wang, D. Liu, J. Yang. 2015. Deep networks for image super-resolution with sparse-prior. IEEE, International conference on computer vision.

[24] J. Kim, J.K. Lee, and K.M. Lee. 2016. Deeply recursive convolutional networks for image super-resolution. IEEE, Conference on computer vision and pattern recognition.

[25] Simonyan, K. Vedaldi, A. Zisserman A. 2013. Deep inside convolutional networks: visualizing image classifications models and saliency maps. arXiv.

[26] Yosinski J, Clune J., Nguyen A, Fuchs T, Lioson H. 2015. Understanding neural networks with deep visualization. arXiv.

[27] Yang C.Y, Ma C, Yang M.H. 2014. Single-image super resolution: a benchmark. Springer, Computer Vision (ECCV) pages 372-386.

[28] Shan Q, Li Z, Jia J, Tang C.K. 2008. Fast image/video upsampling. ACM Transactions on Graphics.

[29] Kim K.I, Kwon Y. 2010. Single image super-resolution using sparse regression and natural image prior. IEEE transactions on Pattern analysis and machine learning, pages 1127-1133.

[30] Xiong Z, Sun X, Wu F. 2010. Robust web image/video super resolution. IEEE transactions on image processing.