Super Resolution survey paper

Xianliang He, Haomin Li, and Qi Chu
{x57he, h439li, q4chu}@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Abstract

Super Resolution is a fundamental and important problem in computer vision. Some promising results have already been achieved in example based enhancement datasets, but the speed and image quality is still far from satisfactory. The performance on Resolution can lead to great benefits in both Computer Vision industry and some real world projects, such as small object detection, renewing old pictures and suspects detection. In this report, we reviewed most of related deep learning research papers on Super Resolution and summarized current progress and potential improvements in the future. We found that most of the effective improvements are made via one of the three approaches: (i) Propose completely new models (ii) Improve the training scheme and loss function to produce better model performance. (iii) Modify the existing model architecture to overcome the model bottlenecks.

Introduction

Image Super-resolution is the process of improving low-resolution images to high-resolution images. It is one of the significant part of computer vision study. The applications range from image compression, medical imaging, security etc. Traditional methods to solve this problem includes bicubic interpolation, Lanczos resampling, statistical image priors etc. As deep learning methods continue to grow rapidly, deep learning based frameworks have been outperformed traditional approaches for image super-resolution task. Before diving into the deep learning approaches, more detailed background of super-resolution task will be introduced.

Problem Definitions

Low-resolution images are obtained from degradation process from high-resolution images. Since the degradation process is unknown, the goal is to model the degradation process and recover the high-resolution images. The degradation can be decomposed into three main operations. The first is certain blur kernel applied on the original input. The second is downsampling operation with some scale factor s. The third is Gaussian white noise.

The objective of the problem usually will be the pixel-wise squared loss between generated high-resolution images and the target high-resolution images with some regularization term.

$$\mathcal{D}(I_y; \delta) = (I_y \otimes \kappa) \downarrow_s + n_\varsigma, \{\kappa, s, \varsigma\} \subset \delta,$$

Figure 1: Formula of the degradation process

Datasets

There are multiple datasets available for image super-resolution problem which differs in resolution, amounts of images etc. Some have both LR and HR images and some only have target images and LR images are obtained from degrading functions, for example, imresize. (Zhihao Wang 2019)

Evaluation Criteria

Peak signal-to-noise Ratio(PSNR) PSNR is the most commonly used metric for reconstruction quality. PSNR is mainly depends on the pixel-wise mean squared loss. The disadvantage of PSNR is that it does not care about human perception of the image. However, due to the short of accurate perceptual metrics, PSNR is still very popular among all the metrics.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(I(i) - \hat{I}(i))^2$$

$$PSNR = 10 * \log_{10}\frac{L^2}{MSE}$$

Structural Similarity(SSIM) Structural similarity aims to capture the structural similarity between images based on luminance, contrast and structure. Luminance is the mean of the pixel values, contrast is the variance of the pixel values and structure is represented as the covariance between two images. This method is also widely used since it considers how human evaluates pictures. Moreover, mean structural similarity(MSSIM) is introduced to consider the locality of image structure. It divides images into few windows and compute the SSIM of each window and averages all the SSIM to obtain the result.

Mean Opinion Score(MOS) This method is a purely subjective method which assigns human images to score quality of images from 1 to 5. The variance and subjective can be largely eliminated as more people involved into the assessment. This method is the most reliable perceptual metrics and some models are poor in PSNR and SSIM but high in MOS.

Related Work

SRCNN

SRCNN (Super-Resolution Convolutional Neural Network) is the first convolutional neural network base method to deal with Single Image Super Resolution. (In contrast, Multi-image super resolution means reconstructing high resolution image through multiple low-resolution images) According to 'Learning a Deep Convolutional Network for Image Super-Resolution', the process of SRCNN is as followed:(Chao Dong and Tang 2014)

1. Preprocessing: Upscale the image to the desired size using bicubic interpolation Y. (lower down the resolution)
2. Feed the image into the network
3. Convert the input to YCbCr color space (since only luminance channel (Y) is used by the network $conv1 \rightarrow relu1 \rightarrow conv2 + relu2 \rightarrow conv3$
4. Merge the output with interpolated CbCr channels to produce the final image.

Figure 2: SRCNN results compared with example based methods

It outperforms the previous example based methods (figure 2) and it only consists of 3 convolutional layers which are:

1. Patch extraction and representation: this operation extracts (overlapping) patches from the low-resolution image Y and represents each patch as a high-dimensional vector. The dimension of the vector represents the set of feature maps of the image. The first layer is expressed as:

$$F_1(Y) = max(0, W_1 * Y + B_1)$$

Where:

W_1 = filters which has size $c * f_1 * f_1 * n_1$
B_1 = biases
c = the number of channels in the input image
f_1 = spatial size of a filter
n_1 = number of filters
B_1 = an n_1 dimensional vector, whose each element is associated with a filter.

2. Non-linear mapping: this operation maps the vector we get from the first convolutional layer to another high-dimensional vector which represents the high-resolution feature maps. . The second layer is expressed as:

$$F_2(Y) = max(0, W_2 * F_1(Y) + B_2)$$

Where:

W_2 = filters which has size $n_1 * 1 * 1 * n_2$
B_2 = an n_2 dimensional vector

3. Reconstruction: this operation aggregates the high resolution patch we get from the second convolutional layer and produce the final high resolution image. The output layer is expressed as:

$$F(Y) = W_3 * F_2(Y) + B_3$$

Where:

W_3 = filters which has size $n_2 * f_3 * f_3 * c$
B_3 = a c dimensional vector

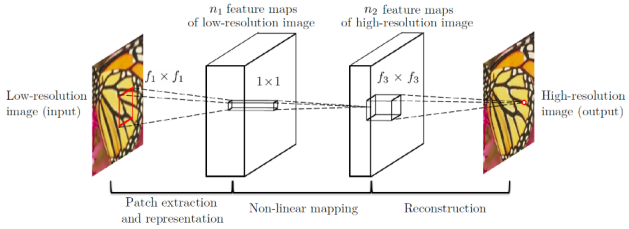

Figure 3: SRCNN visualization

SRCNN has only convolutional layers which provides the advantage that the input image can be of any sizes and algorithm is not patch-based. While the steps look simple, SRCNN is really difficult to train. Hyper parameter changes can greatly affect the SRCNN model and the parameters listed in the paper (learning rate = 10^{-4} for the first two layers, 10^{-5} for the last layer, SGD optimizer) leads PyTorch not to produce the optimal result.

Loss function improvement There are some modifications can be applied to enhance our SRCNN model. Mentioned in the paper, SRCNN used Mean Square Error(MSE) to train the model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} |F(Y_i, \theta) - X_i|^2$$

As we see the result of the original model, the image we get by minimizing MSE is way too smooth since MSE tends to produce an image resembling the mean of all possible high-resolution pictures, resulting in a given low resolution picture. There is another problem of MSE MSE does not capture the perceptual differences between models output and

the ground truth image. For example, We want to preserve the content of the original image, however, the resulting image contains several pixels of the original image shifted to the left. Thus, we replace Mean Square Error (MSE) method by Perceptual Loss. 'Perceptual Losses for Real-Time Style Transfer and Super-Resolution' stated that Perceptual Loss uses high-level image feature representations extracted from the pretrained CNN (Convolutional Neural Network).(Justin Johnson 2016) By replacing the loss function, we can upscale the result to be made up of objects resembling the original one as much as possible since network trained for image classification can store in its feature maps the information on what detailed of common objects look like.

FSRCNN

In 2016, Chao Dong et al. purposed a method called FSRCNN (Fast Super Resolution Convolutional Neural Network) to solve Single Image Super Resolution problem.(Chao Dong 2016) FSRCNN has five following steps: (M: the number of mapping layer, d: LR feature dimension, s: after shrinking , LR feature is reduced from d to s)

1. Feature Extraction: Feed the low resolution image input to the model
2. Shrinking: add a linear combination like filter within the low resolution features. Adopt a small filter number which reduce the low resolutions dimension. This step could significantly reduce the number of parameters.

$$Conv(1, s, d)$$

3. Non-linear mapping: In order to balance between the performance and network scale, we adopt a medium filter size (width) $f_3 = 3$ and use multiple $3 * 3$ layers (depth). With the purpose of being consistent, all mapping layers contain the same number of filter $n_3 = s$. The non-linear mapping part can be represented as

$$M * Conv(3, s, s)$$

4. Expanding: While the second process (Shrinking) cut down the number of low resolution feature dimension to achieve computational efficiency, this step expand the high resolution feature dimension. We adopt $1 * 1$ filters as same as the number of low resolution feature extraction layer.

$$Conv(1, d, s)$$

5. Deconvolution: This step integrate the previous features with a set of deconvolution filter. In terms of deconvolution, the filter is convolved with the image with stride $1/k < 1$, and the output is k times of the input. We adopt 9x9 filters and the deconvolutional layer can be expressed as:

$$DeConv(9, 1, d)$$

In terms of activation function for each layer, we would use PReLU which is very similar to ReLU but can avoid the dead feature caused by zero gradient in ReLU.

$$f(x_i) = max(x_i, 0) + a_i min(x_i, 0)$$

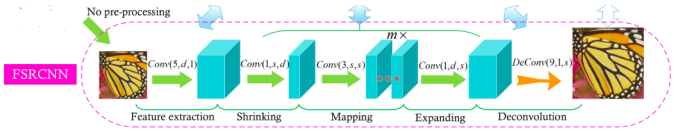

Figure 4: FRCNN visulization

where xi is the input signal of the activation f on the i-the channel and ai is the coefficient of the negative part can be learned by PReLU.

The overall pipeline can be expressed as:
Conv(5,d,1) $\rightarrow PReLU \rightarrow Conv(a, s, d) \rightarrow PReLU \rightarrow m * Conv(3, s, s) \rightarrow PreLU \rightarrow Conv(1, d, s) \rightarrow PreLU \rightarrow DeConv(9, 1, d)$

Compare with SRCNN While SRCNN concept is easy to understand, its processing speed on large image is very slow. Besides, SRCNN required the input to be upsampled to a desired size using bicubic interpolation which result is a quadratically computation complexity with regards to the spatial size of high resolution image. It is not quite efficient. FSRCNN was developed based on SRCNN. Because of not using bicubic interpolation and the modification of the structure, FRCNN achieves a speed up of more than 40x than SRCNN while keeping the similar performance. The computational complexity for SRCNN and FRCNN are as followed respectively:

$$SRCNN : O(f_1^2 * n_1 + n_1 * f_2^2 + n_2 * f_3^2) S_H R$$

$$FRCNN : O(9 * m * s^2 + 2s * d + 106d) S_L R$$

ESPCN

While SRCNN and FSRCNN upscales the image first, then learns from it, ESPCN does not. ESPCN uses a sub-pixel convolutional neural network, is the first convolutional neural network capable of real time super-resolution of 1080p videos on a single K2 GPU. ESPCN offers a significant reduction in computational and memory cost compared to these traditional Super Resolution methods.

There are three major defining characteristics of ESPCN:

- Feature extraction are all executed in LR space with nonlinear convolutions, even allowing the real time super-resolution of HD videos
- ESPCN has $L - 1$ features maps, where the network has L layers, compared to the single upscaling filter with previous methods
- Implicit learns the processing necessary for SR

The methodology of ESPCN revolves around the task of estimating a HR image I^{SR} from a L^R image I^{LR}. To produce the low-resolution image from the original data set of high-resolution images, the original image I^{ORG} is convolved with a Gaussian filter to stimulate the point spread function of the camera. Then the image is down sampled by a factor of r, which then produces I^{LR}. Compared to SRCNN, where it first upscales I^{LR} instead of using the unchanged I^{LR}. In Wenzhe et al.s architecture, an l convolution layer is applied to the untouched I^{LR} and then a sub-pixel convolution layer is added to learn the feature maps

necessary for upscaling. For a network consisting of L layers, the first $L-1$ layers are depicted by the Figure 5. Where $W_l, b_l\ , l \in (1, L1)$ are network weights that are learned, and biased respectively. W_l is a 2D convolution tensor of dimension $n_{l-1} \times n_l \times k_l \times k_l$ where n_l is the number of features at layer l, $n_0 = C$, and k_l is the filter size at layer l. the biases bl are vectors of length n_l. The nonlinearity function ϕ is applied element-wise and is fixed. The last layer f_L converts the low-resolution feature maps to the high-resolution image I^{SR}.

$$f^1(\mathbf{I}^{LR}; W_1, b_1) = \phi\left(W_1 * \mathbf{I}^{LR} + b_1\right),$$
$$f^l(\mathbf{I}^{LR}; W_{1:l}, b_{1:l}) = \phi\left(W_l * f^{l-1}\left(\mathbf{I}^{LR}\right) + b_l\right),$$

Figure 5: Formula of the first $L-1$ layers

Compare with SRCNN and Other Methods SRCNN starts with upscaling and interpolating I^{LR}, which introduces significant computation complexity as CNNs run times are dependent on the input size. Hence ESPCN will implement the approach of increasing the resolution by the very end of the network. This allows the learning process to be done in low resolution. Compared to bicubic filtering and other hand-crafted models, deep neural networks and other CNNs offer increased reconstruction accuracy and higher flexibility. In an ESPCN, it is necessary at some step to increase the resolution or input of the image to a higher resolution. (Wenzhe Shi 2016)

DBPN

DBPN is another neural network that uses an intriguingly novel structure compared to others surveyed in this paper, it characterises itself counterintuitively by using downsampling filters after upsampling. The sampling layers provides an error feedback mechanism for projection errors at each stage and addresses the mutual dependencies of low- and high-resolution images. (Muhammad Haris 2018)

DBPN has the following key features:

1. **Error feedback**: Iterative error correcting feedback mechanism for super sampling, this is done by doing aggregating both up and down projection errors to guide reconstruction.

2. **Mutually connected up- and down-sampling stages**: usual deep learning methods are feed-forward mappings, which is one way and is limited by the richness of the input. This approach is clearly limited by the features in the Low-Resolution space, especially in the case of large scaling factors. Figure 2 (d)

3. **Deep concatenation**: DBPN network has components that represent distinct types of image degradation. These components can be concatenated directly and without propagating them through the network (unlike other feedback algorithms). Figure 2 (d)

4. **Dense connections**: Each up- and down- sampling stage is densely connected to increase feature reuse

Let's look at the execution of DBPN by first explaining the major building block of the DBPN architecture: projection unit. Which can be trained to map either an LR feature map to an HR map or vice versa.

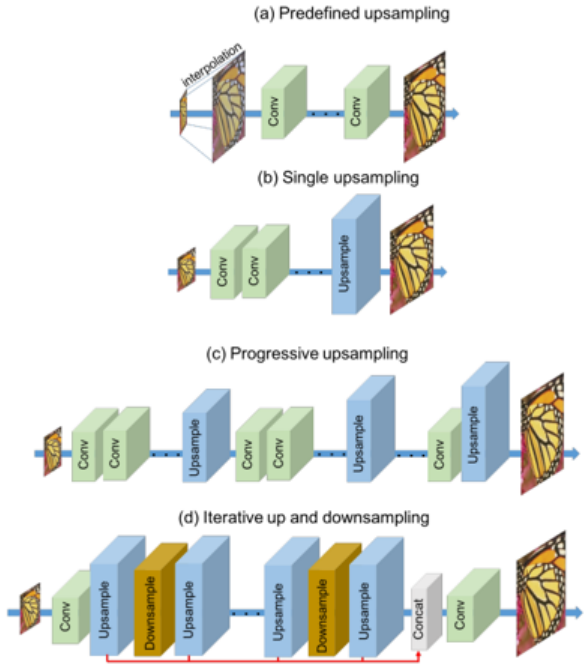

Figure 6: Comparison between different types of sampling and DBPN (d)

Up-projection unit is defined as follows:

scale up:	$H_0^t = (L^{t-1} * p_t) \uparrow_s,$	(1)
scale down:	$L_0^t = (H_0^t * g_t) \downarrow_s,$	(2)
residual:	$e_t^l = L_0^t - L^{t-1},$	(3)
scale residual up:	$H_1^t = (e_t^l * q_t) \uparrow_s,$	(4)
output feature map:	$H^t = H_0^t + H_1^t$	(5)

Figure 7: * is the spatial convolution operator, $\uparrow_s$ is up-sampling operator, $\downarrow_s$ is the down-sampling operator with scaling factor s, p_t, g_t, q_t are (de)convolutional layers at stage t.

The above steps are explained below:

0. Input: a previously computed LR feature map L^{t-1}
1. Map to intermediate HR map H_0^t
2. Map back to LR map L_0^t
3. Take residual difference between L^{t-1} and L_0^t
4. Map residual difference to HR, producing H_1^t
5. Output final HR map from H_0^t (step 2) and H_1^t (step 5)

Down projection unit is quite similar (Figure 8).

The projection units are then organized in the manner below, where an alternation between H and L occurs. This builds a self correcting process that feeds the projection error forward into the sampling layer, which iteratively changes the solution to minimize the error. Moreover, large filter of

scale down: $$L_0^t = (H^t * g_t') \downarrow_s, \quad (6)$$

scale up: $$H_0^t = (L_0^t * p_t') \uparrow_s, \quad (7)$$

residual: $$e_t^h = H_0^t - H^t, \quad (8)$$

scale residual down: $$L_1^t = (e_t^h * g_t') \downarrow_s, \quad (9)$$

output feature map: $$L^t = L_0^t + L_1^t \quad (10)$$

Figure 8: downsampling steps

8 x 8 and 12 x 12 are used in the network. This often yields sub-optimal results but it is subsided by the design of the DBPN network, where the iterative usage of project units suppress this limitation. (Figure 9)

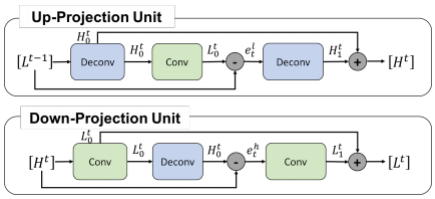

Figure 9: DBPN network consisting of up- and down- projection units

Compare with Other Methods All other surveyed techniques in this paper approach the Super Resolution problem by attempting to learn a non-linear feature map that maps Low-Resolution to High-Resolution. Instead of purely feeding forward, DBPN uses feedback connections to adjust and better the learning process. (Wenzhe Shi 2016)

Residual Learning

Inspired by VGG-net, VDSR is using a much deeper convolutional network than SRCNN.(Jiwon Kim and Lee 2016) This approach is introduced to solve few limitations of SRCNN. The limitations and how VDSR eliminates them will be explained in the following section.

Context SRCNN relies on context of small image regions which would be a problem with high scale factor since small area of information is not sufficient for detail recovery. By increasing the depth of network to 20, the receptive field has been greatly increased. The network is using 3*3 filters and allows (2D+1) * (2D+1) receptive field. By increasing the receptive field, each region of reconstructed image depends on very large region of the low-resolution image.

Convergence Instead of training the low-resolution image to directly construct the target high-resolution image, VDSR trains the residual between target and input images and add up the input and residual to obtain output. The researchers propose to decompose the training process into carrying to the end-layer and reconstructing residual. SRCNN takes too much effort to conduct both tasks, but VDSR only focuses on the second task which fasten the convergence speed. In addition to that, they also increase the training rate. To avoid gradient vanishing problem when learning

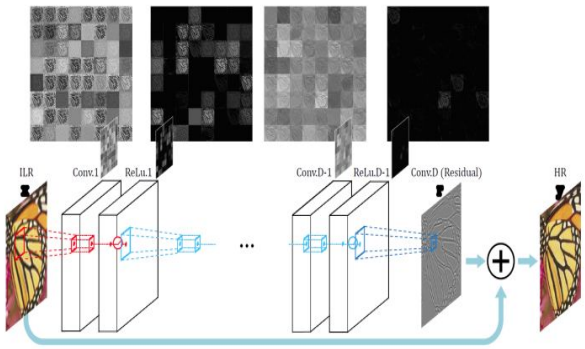

Figure 10: Architecture of VDSR

rate increases.The technique called adjustable gradient clipping is applied. This technique restrict the gradient to be in the range of $-\frac{\theta}{\gamma}$ to $\frac{\theta}{\gamma}$ instead of $-\theta$ to θ. The intuition is to apply heavier clipping on gradient when learning rate is larger. The result shows that the network is much more faster than SRCNN(4 hours vs few days)

Scale SRCNN is trained only for a single scale factor and works on specified scale. However, it is not efficient and scalable to maintain multiple models for different scale factors. Experiments show that if training set is trained on data with scale factor 2 and test set with scale factor 3, then the test performance will be highly degraded. Therefore, the researchers train the network on images with all scale factors and the performance achieves similar performance with single scale factor machines.

Conclusion

In summary, there are overall three approaches to overcome Super-resolution problem. Firstly, developing a new model such as SRCNN, which firstly introduced Convolutional Neural Network to Super-resolution problem. Secondly, modifying an existing model such as FSRCNN which changes the number of Convolutional layers from three to five. Last but not least, making certain improvements in the certain theme, such as replacing the loss function, iteratively down- and up-scaling, as well as shifting the time and place of upscaling. Super-resolution problem has gained more and more attention in recent years. With the rapid development of deep learning, many deep-learning based methods has emerged. We have developed five successful deep-learning methods ranging from first super-resolution framework to complex networks. Although those networks achieve great success, there are still space for improvement. Firstly, the loss and evaluation of super-resolution task has always been a problem. It is still difficult to measure performance of the model in both mathematical and perceptual way. Secondly, efforts will be made to apply solutions to real-world application. For instance, scale factors used for training are limited whereas arbitrary scale need to be handled in practice.

References

Chao Dong, Chen Change Loy, K. H., and Tang, X. 2014. Learning a deep convolutional network for image super-resolution. *ECCV*.

Chao Dong, Chen Change Loy, X. T. 2016. Accelerating the super-resolution convolutional neural network. *ECCV*.

Jiwon Kim, J. K. L., and Lee, K. M. 2016. Accurate image super-resolution using very deep convolutional networks. *CVPR 2016 Oral*.

Justin Johnson, Alexandre Alahi, L. F.-F. 2016. Perceptual losses for real-time style transfer and super-resolution. *arXiv:1603.08155*.

Muhammad Haris, Greg Shakhnarovich, N. U. 2018. Deep back-projection networks for super-resolution. *arXiv:1803.02735*.

Wenzhe Shi, Jose Caballero, F. H.-J. T. A. P. A. R. B. D. R. Z. W. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *arXiv:1609.05158*.

Zhihao Wang, Jian Chen, S. C. H. 2019. Deep learning for image super-resolution: A survey.