Image Super-Resolution Using Deep Convolutional Networks

Amritha S Dinesh S

Abstract – We use a deep learning method for single image super-resolution (SR). The method directly learns an end-to-end mapping between the low/high-resolution images. The mapping is represented as a deep convolutional neural network (CNN) that takes the low-resolution image as the input and outputs the high-resolution one. We further show that traditional sparse-coding-based SR methods can also be viewed as a deep convolutional network while jointly optimising all layers.

Keywords : Super-resolution, deep convolutional neural networks, sparse coding, deblurring

1 Introduction

Recovering a high-resolution image from a single low-resolution image, is a classical problem in computer vision.it is an under determined inverse problem, of which solution is not unique.

The sparse-coding-based method is one of the representative external example-based SR methods. This method involves several steps in its solution pipeline. First, overlapping patches are densely cropped from the input image and pre-processed (e.g.,subtracting mean and normalization). These patches are then encoded by a low-resolution dictionary. The sparse coefficients are passed into a high-resolution dictionary for reconstructing high-resolution patches. The overlapping constructed patches are aggregated (e.g.,by weighted averaging) to produce the final output. This pipeline is shared by most external example-based methods. This pipeline is equivalent to a deep convolution neural network. The entire SR pipeline is fully obtained through learning, with little pre/post-processing.

The model is called Super-Resolution Convolutional Neural Network (SRCNN). It is improved by introducing larger filter size in the non-linear mapping layer. The SRCNN can also process three colours simultaneously.

2 Related Work

2.1 Image Super Resolution

Amongst all the image super resolution algorithms, patch based(example based) achieve state-of-the-art performance. The sparse coding-based method and its improvements are among the state-of-the-art SR methods nowadays. In these methods, the patches are the focus of the optimization; the patch extraction and aggregation steps are considered as pre/post-processing and handled separately.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are regularized versions of multilayer perceptrons. CNNs use relatively little pre-processing compared to other image classification algorithms hence used for natural image denoising. This means that the network learns the filters that in traditional algorithms, were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

2.3 Deep Learning for Image Restoration

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 SRCNN optimizes an end-to-end mapping. Further, the SRCNN is faster at speed. It is not only a quantitatively superior method, but also a practically useful one.

3 CONVOLUTIONAL NEURAL NETWORKS FOR SUPER-RESOLUTION

3.1 Formulation

We wish to learn a mapping F, which conceptually consists of three operations: 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. These vectors comprise a set of feature maps, of which the number equals to the dimensionality of the vectors.

F1(Y) = max (0,W1Y+B1)

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.

F2(Y) = max (0,W2F1(Y) +B2)

3)Reconstruction:this operation aggregates the above high-resolution patch-wise representations to generate the final high-resolution image.

F(Y) =W3F2(Y) +B3.

3.2 Relationship to Sparse-Coding-Based Methods

In the sparse-coding-based methods, let us consider that an f1 × f1 low-resolution patch is extracted from the input image. Then the sparse coding solver, like Feature-Sign , will first project the patch onto a (low-resolution) dictionary. If the dictionary size is n1, this is equivalent to applying n1 linear filters (f1 × f1) on the input image (the mean subtraction is also a linear operation so can be absorbed). The sparse coding solver will then iteratively process then1coefficients. The outputs of this solver are n2 coefficients, and usually n2=n1in the case of sparse coding. These n2 coefficients are the representation of the high-resolution patch. In this sense, the sparse coding solver behaves as a special case of a non-linear mapping operator, whose spatial support is 1 × 1. However, the sparse coding solver is not feed-forward,i.e.,it is an iterative algorithm. On the contrary, our non-linear operator is fully feed-forward and can be computed efficiently. If we set f2= 1, thenour non-linear operator can be considered as a pixel-wise fully-connected layer. It is worth noting that “ the sparse coding solver ” in SRCNN refers to the first two layers,but not just the second layer or the activation function(ReLU). Thus the nonlinear operation in SRCNN is also well optimized through the learning process. The above discussion shows that the sparse-coding-based SR method can be viewed as a kind of convolutional neural network (with a different non-linear mapping). But not all operations have been considered in the optimization in the sparse-coding-based SR methods.On the contrary, in our convolutional neural network,the low-resolution dictionary, high-resolution dictionary,non-linear mapping, together with mean subtraction and averaging, are all involved in the filters to be optimized.So our method optimizes an end-to-end mapping that consists of all operations.

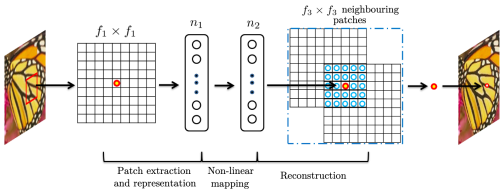


図 1 An illustration of sparse-coding-based methods in the view of a convolutional neural network.

3.3 Training

Learning the end-to-end mapping function F requires the estimation of network parameters Θ=W1,W2,W3,B1,B2,B3. This is achieved through minimizing the loss between the reconstructed images F(Y; Θ)and the corresponding ground truth high-resolution images X. Given a set of high-resolution imagesXi and their corresponding low-resolution images Yi , we use Mean Squared Error (MSE) as the loss function:L(Θ) =1 / n * square(i=1——F(Yi; Θ) − Xi——), where n is the number of training samples. Using MSE as the loss function favors a high PSNR. The PSNR is a widely-used metric for quantitatively evaluating image restoration quality, and is at least partially related to the perceptual quality. It is worth noticing that the convolutional neural networks do not preclude the usage of other kinds of loss functions, if only the loss functions are derivable. If a better perceptually motivated metric is given during training, it is flexible for the network to adapt to that metric. In particular, the weight matrices are updated as

where l ∈ 1,2,3 and i are the indices of layers and iterations, η is the learning rate, and ∂ L/∂ W‘is

$$\Delta_{i+1} = 0.9 \cdot \Delta_i - \eta \cdot \frac{\partial L}{\partial W_i^\ell}, \quad W_{i+1}^\ell = W_i^\ell + \Delta_{i+1}$$

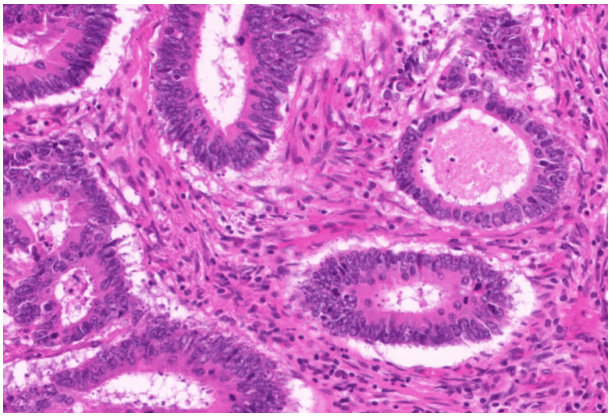
the derivative.The filter weights of each layer are initialized by drawing randomly from a Gaussian distribution with zero mean and standard deviation 0.001 (and 0 for biases). The learning rate is$10 - 4$for the first two layers, and $10 - 5$for the last layer. We empirically find that a smaller learning rate in the last layer is important for the network to converge.

To avoid border effects during training, all the convolutional layers have no padding, and the network produces a smaller output (square(fsub − f1 − f2 − f3+ 3) × c). The MSE loss function is evaluated only by the difference between the central pixels of Xi and the network output.

4 EXPERIMENTS

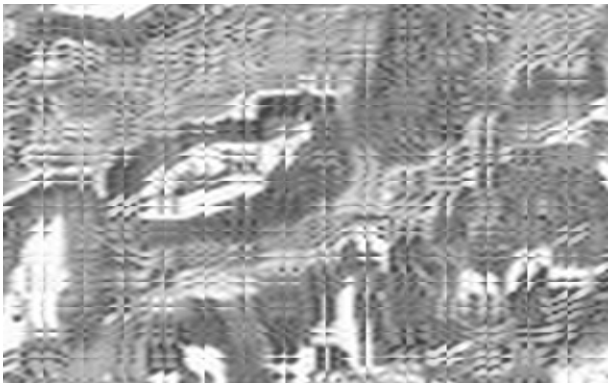
4.1 Training data

For comparison, we use relatively small training set of 70 images. Below is one of the training images.



4.2 Test data and output

We have used 8 images for testing the data one of whose outputs are below.



4.3 Learned Filters for Super-Resolution

Interestingly, each learned filter h as its specific functionality. For instance, the filters g and h are like Laplacian/Gaussian filters, the filters a, e are like edge detectors at different directions, and the filter f is like a texture extractor.

5 Proposed Work

The proposed approach, SRCNN, learns an end-to-end mapping between low and high-resolution images, with little extra pre/post-processing beyond the optimization. SRCNN could be applied to image deblurring and simultaneous SR+ denoising.

The loss is minimized using stochastic gradient descent with the standard backpropagation.

6 References

Image Super-Resolution Using Deep Convolutional Networks : https://arxiv.org/pdf/1501.00092v3.pdf