Learning Visual Co-Occurrence with Auto-Encoder for Image Super-Resolution

Yudong Liang, Jinjun Wang*, Shizhou Zhang and Yihong Gong
Institute of Artificial Intelligence and Robotics, Xi’an Jiaotong University, 28 West Xianning Road, Xi’an, Shaanxi, China, 710049
{liangyudong, shizhouzhang}@stu.xjtu.edu.cn, {jinjun, ygong}@xjtu.edu.cn

Abstract—This paper proposes a novel neural network learning the essential mapping function between the low resolution and high resolution image for Image superresolution problem. In our approach, patch recurrence property of small patches in natural image are utilized as a prior to train the network. An autoencoder neutral network is designed to reconstruct the high resolution patches. The constraint that the output of the coding part should be similar as the corresponding high resolution patches is imposed to ameliorate the illness nature of the superresolution problem. In fact, the degeneration mapping from the high resolution image to the low resolution image is also integrated in the network. Both visual improvements and objective assessments are demonstrated on true images.

I. INTRODUCTION

The image resolution is limited by the imaging sensors and the image quality is deteriorated during image acquisition when camera motions, lens blur and other noise exist. Superresolution constructs high-resolution (HR) images from observed low-resolution (LR) images. Superresolution is an ill-posed problem and has aroused a lot of interests since 1980s. The techniques have varied from frequency domain to spatial domain and mainly contain several categories: interpolation methods [1], reconstruction methods [2][3] and example based methods [4][5][6]. The first important frequency domain method dates back to Tsai and Huang [7]. Although some extensions have been proposed to handle some more complicated situations [8], recent works have almost focused in the spatial domain.

Many approaches have been proposed in spatial domain such as many interpolation methods. Interpolation methods take advantage of the fact that pixels change gradually in local area and continuous events can be estimated from discrete pixels. A widely applied interpolation method is bicubic interpolation method [1], which has a high computational efficiency and can produce smooth and natural results. The bicubic interpolation sometimes fails to capture the sharp changes around edges which leads to blurred edges and annoying artifacts.

Due to the nature of the ill-posed problem, reconstruction methods use image priors to improve the situation. A widely accepted prior, global reconstruction constraint means that resulted HR images can be degenerated to the original LR images [2]. It can be combined with another prior, image gradient prior [3] to further improve the situation.

A recently evolving methodology develops the priors by using examples from other images externally [4] or the image itself internally [9]. The example based work depends on a learned co-occurrence prior [4][5] to determine the correspondence between LR and HR image patches. Appropriate image features are extracted from LR and HR images to build image patch sets. Freeman [4] proposed to use image patches to build a large training set, and to apply a Markov Random Field model to get the local coherent HR images. The idea of manifold learning have motivated a simple but effective model based on Neighbor Embedding. Chang [6] proposed that LR and HR image patches exist in two manifolds and locally there is a correspondence between them.

Inspired by development of recent sparse signal representation, Yang [5][10] figured out relationship among HR signals precisely from their low-dimensional projections. Dictionary pairs are built to be combined with coding coefficients to restore image patches. Wang [11] has proved that the coding coefficients are almost invariant across different scales and have shown some interesting applications [12][13]. The internal patch redundancy has drawn a lot of attentions in recent years [9][14]. Another interesting topic is modeling the relationship of the raw LR images from the HR images [15][16]. Netalae Efrat [16] proves that an accurate blur model is more important than a complex image prior.

Neural network is inspired by the function of biological neurons and nervous systems. It is especially suitable for nonlinear prediction problem. An AutoEncoder tries to get an approximation function \( f_\theta(x) \) to the identity function, in order to make output \( \hat{x} \) be similar with input \( x \). By placing constraints on the network, for example, constraining the number of hidden layers’ units, the interesting structure of data can be discovered.

In this work, we use an autoencoder neural network to incorporate learning the degeneration mapping from HR images to LR images. An autoencoder can encode the input effectively and learn a good feature representation for the input. We use a multilayer autoencoder to encode LR patches representation to HR Patches representation then decode HR Patches representation back to LR patches representation. Without any modification, the encoder part for the autoencoder can be use to generate the HR image patches representation.
II. BASIC NEURAL NETWORK FOR IMAGE SUPERRESOLUTION

Following the example based approaches, a mapping function from LR image patches to HR image patches needs to be learnt. A training set is needed to be built. To get a LR images, bluring and downsampling process are required. Now assuming we have a HR image \( I \) we could get the LR images using

\[
P^L = (I \otimes G) \downarrow \frac{1}{2} .
\]

\( G \) denotes a blur kernel, which can be gaussian kernel, \( \downarrow \frac{1}{2} \) is the downsampling process related with downsampling scale \( S \).

A three layers neural network for image superresolution is designed as the first three layers\((L_1-L_3)\) of fig.1. The neural network can be learned by minimizing the reconstruction error between training data\((HR\) image patches \( p^h \)) and network output \( f_{\theta_1} \). We use the \( W_i \) to denote the weights between the \( ith \) and \((i + 1)th\) layer. The input layer \( L_1 \) which is the first layer, receiving the middle frequency image patches as a input \([4]\), is denoted as \( f_{\theta_1} = p^m \). The hidden layer \( L_2 \) has an activating function \( f_{\theta_2} = f_{\text{active2}}(f_{\theta_1} * W_1) \). Then the third layer \( L_3 \) which is output layer has an activating function \( f_{\theta_3} = f_{\text{active3}}(f_{\theta_2} * W_2) \) to get a final output. There is a bias in each neuron, we make an augmentation of \( w = [w \ 1] \), then it comes \( f_{\theta} = f_{\text{active3}}(\sum_{i=1}^{n} w_i x_i) \).

The overall training process aims to minimize the following objective

\[
\min_{\{W_1, W_2\}} f_{\text{active3}}(f_{\text{active2}}(p^m * W_1) * W_2). \tag{2}
\]

This network use a nonlinear function to model the mapping function from the LR image patches to the HR image patches. Back propagation is used to compute the gradient, and stochastic gradient descent method \([17]\) is use to optimize the object function. The weight is calculated as \( W = W - \beta \text{Grad}_{\text{batch}}(W) \). The \( \beta \) is the learning rate.

Then, we form the image patches as a markov random field\((MRF)\) and images patches having a overlapping area will have a connection with each other \([4]\). \( P_i \) denotes the real value of \( ith \) patch and \( p^\text{observed} \) is the observed value of \( P_i \). \( p^\text{observed} \) may contains noises and may be inaccurate. The overlapped patches may have a data discrepancy when restored. Besides, patches should have a local coherence and tend to be smooth. We try to recover the the real value \( P_i \) from the observed value of \( P^\text{observed} \). Belief propagation or graphcut can be applied for the MRF to get a solution. Alternatively, the LLE method \([6]\) alleviates the data discrepancy by simply averaging the overlapping pixels. We average the overlapping pixels for its simplicity in our work.

III. AUTOENCODER NEURAL NETWORK

An autoencoder neural network is an unsupervised learning algorithm. By placing constraints on the network, we could discover interesting structures about the data. We design a five layers autoencoder and the first three layers has the same structure as the basic neural network. This part calls an encoder part. It encodes image patches of the middle frequency layer to image patches of the high frequency layer information. Then the last 3 layers decodes the high frequency layer information back to the middle frequency layer information. it is easily found that the degeneration mapping between HR image patches and LR image patches has been learned from the autoencoder training. And it is natural to constrain the output of the third layer to be similar with high frequency image patches information.

We design the AutoEncoder network as 1. \( L_i \) is denoted as the \( ith \) layer of autoencoder neural network, \( f_{\theta_i} \) as the output of the \( ith \) layer. \( f_{\text{active3}} \) is the activation function of the \( ith \) layer. We have to make a compromise: first, the output of the input should be similar with the input, which learns a good representation for the input and incorporating the degeneration mapping information such as blur model; second, get a good superresolution mapping function from LR image patches to HR image patches which constrains the output of the third layer. Overall, we minimize the following objective function:

\[
||f_{\theta_3}(p^m) - p^m||^2 + \alpha ||f_{\theta_3}(p^m) - p^h||^2 \tag{3}
\]

Parameter \( \alpha \) is a parameter to control the tendency for the learning. Empirically, \( \alpha \) is not so sensitive and we set \( \alpha = 10 \). Empirically, we find our constraints to the output of the third layer are quite powerful. Even without pretraining, the net can converge quite well. The constraints amend the accumulated errors and avoid the backpropagated errors being minuscule. Talking loosely, training the constrained autoencoder can be treated as training two separated three layers neuron network, which effectively avoids the training problem.

The computational load can be analyzed at two stages: training stage and testing stage. At training stage, a five layers autoencoder is trained. The main costs are backpropagation process to calculate the gradients. The computational load varies with the number of the training set and the number of neurons used in each layer. We prepare training data from the ”lenna” picture and restrict the number of 24th and 4th layer, So the training process is quite efficient compared with sparse coding algorithms. At the testing stage, only feedforword
calculation is needed to give the superresolution results. So it is quite fast and efficient at the testing stage.

IV. EXPERIMENTAL RESULTS

In this section, we evaluate our algorithm on natural images which come from Berkeley Segmentation Dataset and Benchmark500(BSDS500) [18]. We take the most commonly used assessment: Peak Signal to Noise Ratio(PSNR) and Structural Similarity(SSIM) [19].

We compare our AutoEncoder approach with the 3 layers Neural network(denoted as NN3,described in section II), Sparse coding methods [10](denotes as ScSR(512) and ScSR(1024),with a coding dictionary size of 512 and 1024 respectively), baseline methods bicubic interpolation method. All the 500 images of BSDS500 are downscaled to \( \frac{n}{scale} \) and then superresolution to the original size. The PSNR and SSIM results are reported in table.I and table.II. The \( \otimes i \) in tables means the magnification factor is \( i \) times.

In table.I and table.II, our methods outperform all the other methods except the PSNR value of \( \otimes 2 \) case. ScSR methods have only slightly better PSNR values when the size of image is doubled and the gap can be ignored. And our methods have a better SSIM evaluation. For the baseline method, Bicubic interpolation method has already got a good PSNR and SSIM evaluation in \( \otimes 2 \) case. Therefore, the result of \( \otimes 3 \) and \( \otimes 4 \) case is more convincing to compare the methods. And our method has constantly better PSNR and SSIM values in these cases. And it is more challenging in these cases as most algorithms degrade rapidly as the magnification factor becomes larger. Besides, our algorithm is more efficient than sparse coding methods. Our method doesn’t have the heavy computational burden for calculating the coding coefficients, the calculating time for the mapping from the LR patches to HR patches can almost be ignored. Besides, the auto encoder approach outperform the 3 layers Neural network which validates that incorporating the modeling of degeneration mapping from HR image patches to LR image patches does help to ameliorate the illness nature of the superresolution problem and get better results. Our method gets visually sharper results and performs especially better in texture-rich regions.

**TABLE I**

<table>
<thead>
<tr>
<th>( \otimes )</th>
<th>AE</th>
<th>NN3</th>
<th>ScSR(512)</th>
<th>ScSR(1024)</th>
<th>Bicubic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>30.6863</td>
<td>30.3992</td>
<td>30.0931</td>
<td>30.7647</td>
<td>29.8711</td>
</tr>
<tr>
<td>3</td>
<td>27.5992</td>
<td>27.7972</td>
<td>27.3897</td>
<td>27.9061</td>
<td>27.6127</td>
</tr>
</tbody>
</table>

**TABLE II**

<table>
<thead>
<tr>
<th>( \otimes )</th>
<th>AE</th>
<th>NN3</th>
<th>ScSR(512)</th>
<th>ScSR(1024)</th>
<th>bicubic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.8715</td>
<td>0.8620</td>
<td>0.8677</td>
<td>0.8689</td>
<td>0.8481</td>
</tr>
<tr>
<td>3</td>
<td>0.769</td>
<td>0.757</td>
<td>0.7477</td>
<td>0.7476</td>
<td>0.7409</td>
</tr>
<tr>
<td>4</td>
<td>0.6945</td>
<td>0.6797</td>
<td>0.665</td>
<td>0.6657</td>
<td>0.6686</td>
</tr>
</tbody>
</table>

It is easily found that PSNR is related to Mean square error(MSE), and Root Mean Square(RMS). PSNR doesn’t always agree with human subjective assessments. We show some pictures to give a illustration for comparison in figure. 2 3 4. From left to right,we show the results corresponding to aforementioned methods: autoencoder, bicubic, NN3, ScSR(512) and ScSR(1024). Our method always give sharper results such as edges of the objects and more clear pictures such as the girl’s face. The fur of the animal which has a rich texture appears more clear and natural. Our method performs especially better when the magnification factor becomes larger. The results of NN3 method are almost as good as sparse coding methods. The incorporating of degeneration mapping has led to our better performance than NN3 methods.

V. CONCLUSIONS

We have designed a constrained autoencoder neural network for superresolution issue which proves to be both effective and efficient. We follow the example based approaches and utilize our network to rebuild HR image patches. Our autoencoder has modeled the degeneration mapping from the HR image patches to LR image patches and its coding part can be used to accomplish the superresolution process. Our superresolution method has demonstrated a sharper results and better outcomes in rich texture regions. Some further extensions such as combining Internal and External training image patches can be investigated in our future work.

ACKNOWLEDGMENT

The authors would like to thank Shun Zhang and Xiaoyu Tao for their insightful comments on the paper. This work is supported by the Open Projects Program of National Laboratory of Pattern Recognition.

REFERENCES

Fig. 2. Comparison of Super-resolution results with a magnification factors 2

Fig. 3. Comparison of Super-resolution results with a magnification factors 3

Fig. 4. Comparison of Super-resolution results with a magnification factors 4


