CNN BASED RETINAL IMAGE UPSCALING USING ZERO COMPONENT ANALYSIS

Andrey Nasonov, Konstantin Chesnakov, Andrey Krylov

Laboratory of Mathematical Methods of Image Processing,
Faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University,
119991, Russia, Moscow, Leninskie Gory, MSU BMK
(nasonov, chesnakov, kryl)@cs.msu.ru

KEY WORDS: Image upscaling, CNN, ZCA Whitening, Retinal Images

ABSTRACT:

The aim of the paper is to obtain high quality of image upscaling for noisy images that are typical in medical image processing. A new training scenario for convolutional neural network based image upscaling method is proposed. Its main idea is a novel dataset preparation method for deep learning. The dataset contains pairs of noisy low-resolution images and corresponding noiseless high-resolution images. To achieve better results at edges and textured areas, Zero Component Analysis is applied to these images.
The upscaling results are compared with other state-of-the-art methods like DCCI, SI-3 and SRCNN on noisy medical ophthalmological images. Objective evaluation of the results confirms high quality of the proposed method. Visual analysis shows that fine details and structures like blood vessels are preserved, noise level is reduced and no artifacts or non-existing details are added. These properties are essential in retinal diagnosis establishment, so the proposed algorithm is recommended to be used in real medical applications.

1. INTRODUCTION

Image upscaling is an important problem for wide range of practical applications where input images are usually noisy such as medical image processing tools, surveillance and satellite systems and many others. In these cases, the development of image resampling methods becomes a challenging task.

Single image upscaling is an underdetermined problem since multiple solutions exist for the input low-resolution image. Thus, different constrains are essential for reducing the diversity of outputs. Due to importance of preserving edges and other high-frequency details for the solution, we wish to save sharpness of non-smooth regions on the output image. Also the image interpolation method should be able to cope with noisy images. Upscaling results should be consistent to noiseless images and images with different levels of noise.

Since precise edge reconstruction is important for human visual perception, the image upscaling algorithms should preserve non-smooth regions in the image and should not add artifacts like ringing effect or blur.

Classical general purpose linear upscaling methods such as bilinear, bicubic, and Lanczos interpolation are fast and work well with noisy images, but they do not perform edge-directed interpolation and produce blurry results.

Edge-directional image upscaling algorithms use the information about image edges to produce an adaptive image interpolation kernel at each pixel. Algorithms EGII (Zhang and Wu, 2006), ICBI (Giachetti and Asuni, 2011) and DCCI (Zhou et al., 2012) use a combination of two directional kernels for pixel interpolation depending on the directions of edges in this pixel. They work well for straight and diagonal edges, but fail at image corners, textured regions with multiple directions and noisy areas.

Modern image upscaling methods are mostly learning-based algorithms that learn a mapping transform between high-resolution and corresponding low-resolution images (patches). Algorithm (Li and Orchard, 2001) obtains this transform individually at each pixel from a self-similarity property of natural images at different scales. The method family SI (Choi and Kim, 2015) puts the patch into one of 625 classes and uses individual interpolation kernel for each class. Convolutional neural network (CNN) methods that directly learn end-to-end transformation between low- and high-resolution images also use the mapping functions. In methods such as SRCNN (Dong et al., 2014) all data of convolutional layers is fully obtained through learning with little pre/post processing. The quality of resulting high-resolution images mostly depends on sufficiency of the training dataset, matching the input image class with image classes from the training data, and effectiveness of CNN coefficient optimization.

Despite the fact that recent CNN models have reported outstanding results, they greatly depend on the training data. If the training data is not sufficient, the results may become unstable: small changes in the input image may result in significant changes in output images. For example, if the image resampling algorithm has been trained using high-quality image set, it will try to recover the details from noise in case of noisy input image. This effect is strongly unwanted for medical image processing where generation of non-existing structures may produce incorrect diagnosis. It also results in noise amplification that is highly noticeable in video resampling.

The proposed algorithm is based on SRCNN model, but it differs in learning dataset preparation. Firstly, we use the training set that consists of natural images with different levels of Gaussian noise for learning process, as in (Nasonov et al., 2016). This leads to accuracy image resampling for both noiseless test examples and noisy natural images including medical images. Secondly, we apply zero component analysis (ZCA) transformation (Krizhevsky, 2009) to training image set. It enhances important details such as edges and textures and also helps to remove unwanted high-frequency noise.

2. PROPOSED METHOD

2.1 Network model

In our method we use the same model for CNN as described in SRCNN algorithm (Dong et al., 2014). Firstly, the input image is upscaled via bicubic interpolation with the certain scale factor. Then that we apply three convolutional filters with non-linear function, which produce the mapping to high-resolution image. The first layer of this function is a convolution of the input image with a filter W_1 of size $9 \times 9 \times 64$ plus bias B_1 and an application of rectified linear unit (ReLU) after the convolution. Here input is a grayscale image and B_1 is a vector of size 64. In other words, on the first layer we apply 64 convolutions with 9×9 sized filters.

Let Y be the low frequency image which is magnified by the bicubic method. Then the first layer is calculated as

$$F_1(Y) = MAX(0, W_1 * Y + B_1). \tag{1}$$

The first layer extracts low-level structures such as edges of different orientations from the low resolution image.

On the second layer we apply ReLU to the convolution of $F_1(Y)$ with a filter W_2 of size $64 \times 5 \times 5 \times 32$ plus bias $B2$, here $B2$ is 32-dimensional:

$$F_2(Y) = MAX(0, W_2 * F_1(Y) + B_2). \tag{2}$$

It maps the features extracted at the previous step from the low resolution sub-space with corresponding features from the high resolution sub-space.

The third layer is used for image reconstruction. It acts like a weighed averaging of the high-resolution feature patches to single pixel. It is a convolution with $32 \times 5 \times 5$ dimensional filter W_3 plus bias B_3:

$$F_3(Y) = W_3 * F_2(Y) + B_3. \tag{3}$$

All these operations form a convolutional neural network, which we name $F(Y, \Theta)$, where Θ is the network filter and bias coefficients.

2.2 Training method

We have to find the network parameters Θ producing the appropriate result for images with and without noise. This is achieved through minimizing the loss between reconstructed images $F(Y, \Theta)$ and ground truth high resolution images. A training image set containing high-resolution images $\{X_i\}$ and their matching low-resolution images $\{Y_i\}$ is used.

We use Mean Squared Error (MSE) as the loss function:

$$L(\Theta) = \sum_{i=1}^{N} \|F(Y_i, \Theta) - X_i\|^2 \tag{4}$$

This leads to higher PSNR values as an objective metric. PSNR is a widely-used metric for quantitatively evaluating image interpolation quality. Though it has weak correlation with human perception, the minimization of the MSE-based loss function produces satisfactory upscaling results, even if they are assessed using other objective metrics, e.g., SSIM, MSSIM.

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

Experiments have shown that using only high-quality images in training dataset leads to noise amplification in noisy testing images. Thus, we modify the training set in order to obtain satisfactory results for noisy low-resolution images. Let name $\{X_i\}$ the images with a lot of high-frequency information from the target image class, which are taken as the ground truth images. In order to make the algorithm effective and stable to noisy input images, we use the following method to generate low-resolution images for scale factor s:

1. Add Gaussian noise with standard deviation σ_n.

2. Apply Gaussian filter with radius $\sigma_s = \sigma_0\sqrt{s^2 - 1}$ for aliasing suppression, $\sigma_0 = 0.3$.

3. Perform image decimation by taking each s-th pixel.

This process can be formulated as:

$$[Y_i]_{x,y} = [(X_i + n_{\sigma_n}) * G_{\sigma_s}]_{sx,sy}. \tag{5}$$

2.3 Zero Component Analysis

As preserving edges and textures is highly desirable in image upscaling task for medical imaging, we want to build a mapping function that minimizes visually salient errors along edges, especially in case of noisy input. Thus, we use Zero Component Analysis (ZCA) transformation (Krizhevsky, 2009) to enhance high-frequency components, such as edges on the image by normalizing the variance of data.

To find a covariance matrix we need to extract patches from the images of the training set and make these patches have zero mean. Assuming normalized image patches x_i are stored as column vectors, the covariance matrix is computed as follows:

$$\Sigma = \frac{1}{m}(XX^T) = \frac{1}{m}\sum_{i=1}^{m}(x^i)(x^i)^T \tag{6}$$

Here X is n-by-m matrix, where i-th column is vectorized i-th image patch, m is number of image patches cut from the image, n is the number of pixels in one patch. Then, we can compute the eigenvectors $u_1, u_2, \ldots, u_n$ and corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the covariance matrix Σ. Here the first eigenvector u_1 is the principal direction of variation of the data, u_2 is the secondary direction of variation, *etc*. These vectors form a new basis in which we can represent the data. Using these eigenvectors as columns, matrix U can be constructed. By multiplying U by vectorized data patch x we will rotate the original data: $x_{rot} = Ux$. To make each of the input features have unit variance, we rescale each feature of the rotated data vector: $x_{\text{PCA}_{\text{white}},i} = \frac{x_{rot,i}}{\sqrt{\lambda_i}}, i = 1, \ldots, n$.

It is important to note that during the evaluation of $x_{\text{PCA}_{\text{white}},i}$, some of the eigenvalues λ_i may be close to 0. As a consequence, the division by $\sqrt{\lambda_i}$ may lead to unstable results. Thus, a regularisation term ϵ is added to the eigenvalues before the division. In our experiments ϵ is set to 0.1.

Finally, ZCA whitening transformation for vectorized patch x is defined as:

$$x_{\text{ZCA}_{\text{white}}} = W_{\text{ZCA}_{\text{white}}} x = U \begin{pmatrix} \frac{1}{\sqrt{\lambda_1+\epsilon}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2+\epsilon}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\sqrt{\lambda_n+\epsilon}} \end{pmatrix} U^T x \quad (7)$$

2.4 Preprocessing with ZCA

ZCA whitening transformation is applied to whole images from the training database in the following way. At first, $k \times k$–sized patches are extracted from the images. After that we calculate mean value, store it and extract it from the patch to make it zero-mean, then ZCA transformation is applied, the old mean value is restored, and central pixel of the patch is copied to its position on the new image.

The edges and other high-frequency details on the resulted images are enhanced after whitening. For ZCA matrix evaluation we use data set consisting of high resolution natural images with a lot of high-frequency infromation. Applying ZCA whitening transformation for the training image database improves the accuracy of the interpolation at edges and textured areas, while adding noise to low-resolution training set suppresses unwanted high-frequency noise in the result.

3. EXPERIMENTS

The performance of the proposed algorithm has been analyzed for the upscaling factor of 2. For objective evaluation, we compare our algorithm with other state-of-the-art methods: original SRCNN (Dong et al., 2014), modified CNN resampling method with noisy training set (Nasonov et al., 2016), DCCI (Zhou et al., 2012), SI-3 (Choi and Kim, 2015) and bicubic interpolation.

3.1 Training data

Original SRCNN algorithm with 9-1-5 configuration of the network was trained by its authors on 91 high-resolution images. We have used a collection of 124 photographic images of nature, buildings and humans (WebShots Premium Collections, October 2007) with average resolution 1600×1200 as a high-resolution training image set for modified CNN resampling method with noisy training set, as described by its authors. The same image set has been used for training SI-3 algorithm.

For the proposed algorithm, ZCA matrix also has been calculated using the same image set. For its calculations we use only noiseless high-resolution reference images, because they represent the desired output. After that, the computed ZCA transformation is applied to high-resolution reference images in the way described in the Section 2.3. Further, following (Nasonov et al., 2016), adding noise and downscaling procedure have been applied to these images using the method to generate low-resolution images using (5). σ_n in this equation is set to 6. These pairs of high- and low-resolution ZCA'd images constitute a new training set for the proposed algorithm. From the training set pairs of patches of the size 32×32 from the low-resolution image and its corresponding central 20×20 (due to the border effects during convolutions) from the high-resolution image are extracted with the stride of 21. For the training procedure we used Caffe package (Jia et al., 2014).

3.2 Learned Filters

Figures 1, 2 shows examples of first layers filters trained on the WebShots Premium Collections database using the proposed and (Nasonov et al., 2016) method.

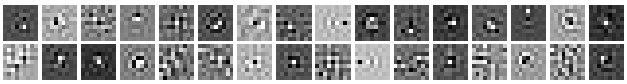


Figure 1. Examples of filters obtained with the proposed learning method

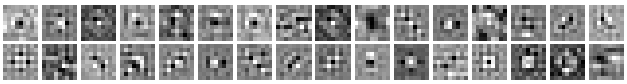


Figure 2. Examples of filters obtained with learning method on noisy training set (Nasonov et al., 2016)

It is clearly seen that both methods have been trained using noisy image set, due to a presence of various noise detection filters. Also filters of texture and edge detectors are presented, but at the same time these filters are adapted for these noisy features. However, even the directional features for the proposed method turned out to be more complicated due to the application of ZCA whitening transformation. It helps the proposed method detect more tricky features on the image and process them in a correct way.

3.3 Testing

We have checked our algorithm on DRIVE database (Staal et al., 2004), consisting of 20 retinal images. These images are downsampled with factor of 2 following (5), then upsampled using image resampling algorithms being tested. The results are compared with reference images using objective metrics PSNR and SSIM (Wang et al., 2004).

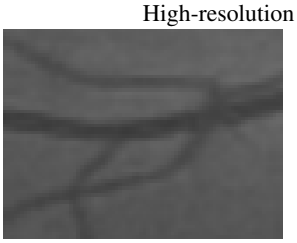
The results of different upscaling methods applied to test retinal images are shown on the Figures 3, 4, 5, 6, 7. It can be seen that the results of bicubic and DCCI (directional bicubic) algorithms are over-smoothed, which is undesirable in ophthalmological image resampling. At the same time, results obtained with original SRCNN algorithm produce instable noisy result, which is highly unwanted in ophthalmological image processing.

On the contrary, the proposed method produces good results from two points of view: it preserves even small vessels in retinal images removing noisy components at the same time. The cause of this effect lies in simultaneously applying ZCA whitening transformation and usage of training dataset containing noisy low-resolution image patches with corresponding noiseless high-resolution patches during learning the kernels of the neural network.

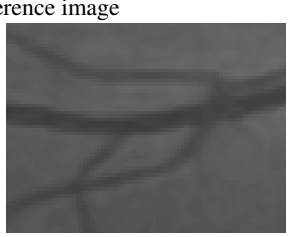
4. CONCLUSION

A novel model for CNN-based upscaling algorithm has been proposed. It learns the mapping transform using noisy low-resolution and reference high-resolution images with preprocessing via Zero Component Analysis. It has been tested on images

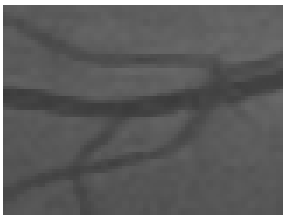
High-resolution reference image



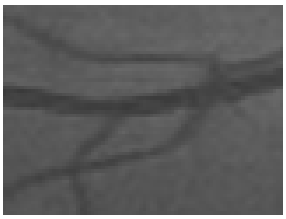
Bicubic result



DCCI result

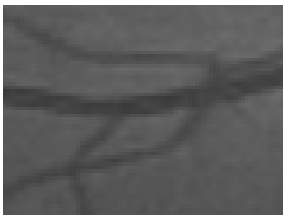


SI-3 result

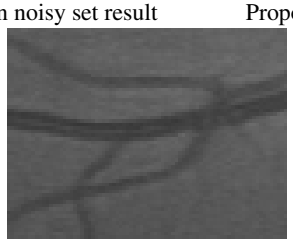


SRCNN result

SRCNN trained on noisy set result



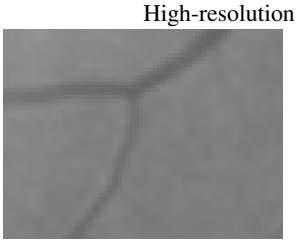
Proposed method



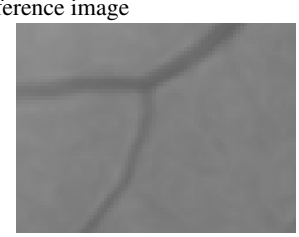
Reference image

Figure 3. The result of retinal image upscaling.

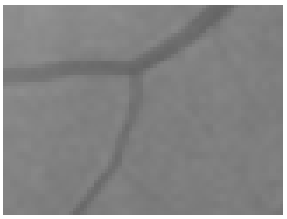
High-resolution reference image



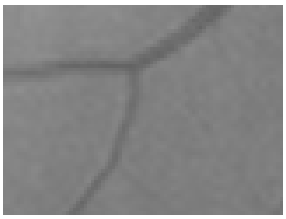
Bicubic result



DCCI result

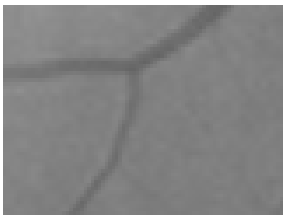


SI-3 result

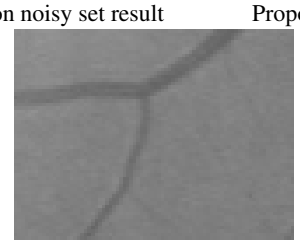


SRCNN result

SRCNN trained on noisy set result



Proposed method



Reference image

Figure 4. The result of retinal image upscaling.

from retinal image database and has shown both high subjective and objective quality. Absence of artifacts and feature preserving make the proposed algorithm a good choice to be used in retinal image processing.

5. ACKNOWLEDGEMENTS

The work was supported by RFBR grant 15-29-03896.

REFERENCES

Choi, J.-S. and Kim, M., 2015. Super-interpolation with edge-orientation based mapping kernels for low complex 2x upscaling. *IEEE Transactions on Image Processing* 25(1), pp. 469–483.

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

Giachetti, A. and Asuni, N., 2011. Real time artifact-free image

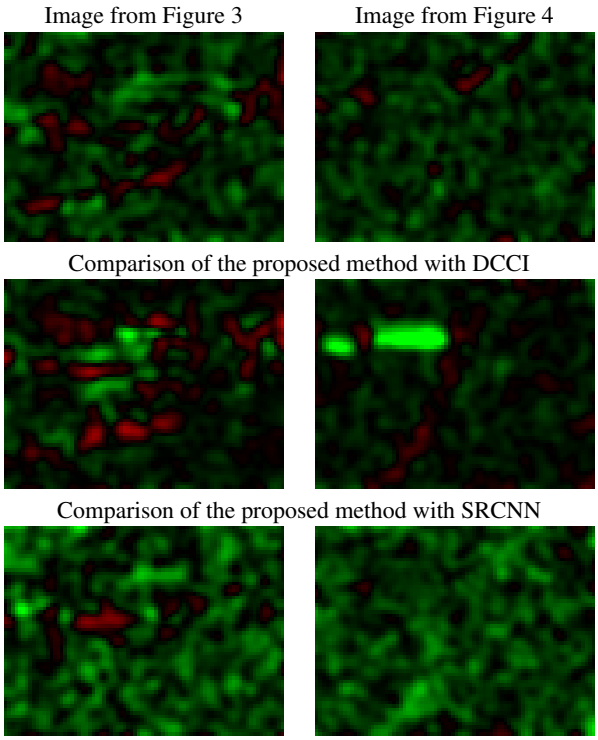


Figure 5. The difference maps between the proposed method and competitor algorithms showing relative local absolute difference with the reference image. Green areas are areas where the proposed method show better results while red — worse.

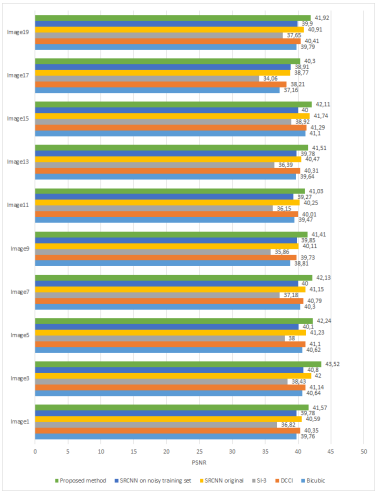


Figure 6. PSNR results for images with odd numbers from DRIVE database

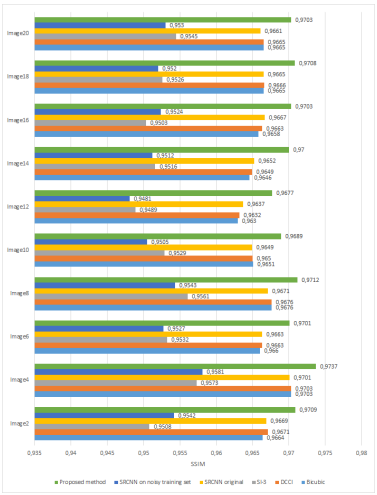


Figure 7. SSIM results for images with even numbers from DRIVE database

interpolation. *IEEE Transaction on Image Processing* 20(10), pp. 2760–2768.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Krizhevsky, A., 2009. Masters Thesis "Learning multiple layers of features from tiny images". `www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf` (8 Apr. 1999).

Li, X. and Orchard, M., 2001. New edge-directed interpolation. *IEEE Transactions on Image Processing* 10, pp. 1521–1527.

Nasonov, A., Chesnakov, K. and Krylov, A., 2016. Convolutional neural networks based image resampling with noisy training set. In: *International Conference on Signal Processing (ICSP2016)*, Chengdu, China, pp. 62–66.

Staal, J., Abramoff, M., Niemeijer, M., Viergever, M. and van Ginneken, B., 2004. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*.

Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4), pp. 600–612.

Zhang, L. and Wu, X., 2006. An edge-guide image interpolation via directional filtering and data fusion. *IEEE Transactions on Image Processing* 15, pp. 2226–2235.

Zhou, D., Shen, X. and Dong, W., 2012. Image zooming using directional cubic convolution interpolation. *IET Image Processing* 6(6), pp. 627–634.