Convolutional neural networks based image resampling with noisy training set

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, MGU BMK
Email: kryl@cs.msu.ru

***Abstract*—A new learning model for image resampling with convolutional neural network is proposed. Its main idea is the dataset preparation method for deep learning. The proposed algorithm can work with noisy and noiseless images and provides good quality for wide noise level range. The method was tested using standard datasets and was also applied for retinal image resampling.**
Keywords: image resampling, convolutional neural networks

I. Introduction

Image interpolation is a classical problem of image processing. Its goal is to recover a high-resolution image from a single low resolution image. Image interpolation algorithms are widely used in a large set of devices and applications, such as medical image processing tools, where the input is often a noisy low resolution image.

This problem is ill-posed since multiple solutions exist for a given low-resolution image. So it is an underdetermined inverse problem. Such problems are typically solved by adding constraints based on prior information.

The simplest algorithms are general purpose linear interpolation methods like bilinear, bicubic, and Lanczos interpolation. These methods represent the neighborhood of each pixel as a polynomial function [1]. They are very fast but do not perform edge-directed interpolation. Usually, a combination of artifacts (blur, ringing and aliasing) comes with linear interpolation.

A large class of edge-directional image interpolation algorithms use prior information about the images. These algorithms assume that the edges look similar at different resolutions. The method [2] improves linear interpolation by kernel elongation along edges. It produces excellent results at straight edges but does not perform well for image corners and textured areas containing multiple directions. Algorithms EGII [3], ICBI [4] and DCCI [5] consider two directions at each pixel using a combination of two directional interpolation results. Algorithm NEDI [6] uses self-similarity of the natural images at different scales to calculate the interpolation coefficients. It works well at both edges and corners but corrupts the structures without self-similarity.

Regularization-based algorithms represent the image resampling problem as a computationally expensive functional minimization problem with data-fitting and stabilizer terms [7]–[9]. The stabilizer holds prior information about the image. Total variation stabilizer is usually used. It smooths the image while keeping the edges sharp.

Example-based strategy is used to find the image constraints that cannot be easily expressed by mathematical models. These methods use internal similarities in the same image or learn mapping functions from the patches from low- and high-resolution training images [10], [11]. The method family SI [12] puts the 3×3 patch into one of 625 classes and uses individual interpolation kernel for each class.

Convolutional neural network that directly learns end-to-end conversion between low- and high-resolution images [13] also uses the mapping functions. In these methods all data of convolutional layers is fully obtained through learning with little pre/post processing. During image upscaling these methods don't need to solve any optimization problems. Quality of the resulted high-resolution images depends on the convolutional model scale, neural network size and the number of iterations during learning process.

The result of example-based image upscaling method greatly depends on the training data. If the training data is not sufficient, the result may become unstable: small changes in the input image may result in high changes in the output image. 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 the appearance of non-existing structures may produce incorrect diagnosis. It also results in noise amplification that is highly noticeable in video resampling.

In this work, we create a new learning model for image resampling with convolutional neural network. The main idea of the proposed model is the dataset preparation method for deep learning. Simple taking noisy high-resolution images as the reference images in the training set result in strong instability and poor quality of the interpolated images. In our approach noise is added only to low-resolution images. It results in resampling quality improvement as well as noise reduction.

II. Proposed method

A. Network model

We use the same model for CNN as described in SR-CNN method [13]. In this scheme, considering a single low-resolution image, we first upscale it to the desired size using bicubic interpolation, but we can still consider this image a low-resolution image. After that we apply 3 layers, where the first layer is a convolution of the input image with a filter of size $9\times9\times64$ plus bias B_1 and an application of rectified linear unit (ReLU) to this 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. On the second layer we apply ReLU to the convolution of first layer's results with a filter of size $64 \times 5 \times 5 \times 32$ plus bias B_2, here B_2 is 32-dimensional. The third layer is used for image reconstruction. It is a convolution with $32 \times 5 \times 5$ dimensional filter plus bias B_3. All these operations form a convolutional neural network, which we name $F(Y, \Theta)$, where Y is a low-resolution input image, Θ is the network filters' and biases' coefficients. So our goal is to recover a high-resolution image from the low-resolution image using this network and the result should be consistent to the noiseless images and images with different levels of Gaussian noise.

B. 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. To get higher PSNR values as an objective metric, we use Mean Squared Error (MSE) as the loss function:

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

Images with a lot of high-frequency information from the target image class are taken as the ground truth images $\{X_i\}$. 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 n-th pixel.

This process can be formulated as:

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

III. Experiments

Experiments have been performed for image resampling with a factor of 2. The proposed method has been compared with state-of-the-art image resampling algorithms: A+ [11], original SRCNN [13], DCCI [5], SI-3 [12] and bicubic interpolation. We have used a collection of 124 photographic images of nature, buildings and humans (WebShots Premium Collections, October 2007) as the high-resolution training image set for the proposed method with noise level $\sigma_n = 8$. The same image set has been used to train SI-3 algorithm. The method A+ has been used as provided by its authors without changes.

The sets Set5 and Set14 [13] have been used to demonstrate the effectiveness of the proposed method. Gaussian noise has been added to each image. The standard deviation of the noise is increased uniformly from 0 to 10 with step 0.1.

The dependence of algorithm performance against noise level is shown in Fig. 4 and Fig. 5. It can be seen that original SRCNN algorithm trained on noiseless images produces low quality results for noisy input images while the proposed method outperforms state-of-the-art algorithms for both noisy and noiseless images.

The Fig. 1 contains the average PSNR and SSIM results for images from the test datasets for multiple noise levels.

The resampling results for 'clown' image with noise $\sigma = 3.4$ are shown in Fig. 2. It can be seen that the proposed algorithm effectively increases the image resolution. At the same time, noise suppression is observed. The cause of this effect is the use of training dataset containing noisy low-resolution image patches with corresponding noiseless high-resolution patches.

A. Retinal image upscaling

The CNN-based approach looks very important to be used for retinal image processing and analysis. The problem of retinal image resampling is often an important preprocessing problem in order to make the evaluation consistent with images in different sizes like in [14] where bicubic interpolation method is used. And the diagnosis results can depend seriously on the resampling method. Below in Fig. 3 two examples of the retina resampling illustrate the difference of the CNN-based and bicubic methods.

It is necessary to mention that the CNN-based retinal image analysis algorithms like [15] started to outperform other classification algorithms. It looks promising to use the CNN-based resampling approach within or together with these algorithms.

IV. Conclusion

The proposed algorithm performs high quality image upscaling as well as effective noise suppression. It looks very promising to be used for medical image resampling within CNN-based medical image analysis. The proposed algorithm gives the best result compared to other resampling methods but at the same time it is not fast enough for realtime applications. It is about 10 times slower than other resampling methods using mapping functions. Nevertheless, it can be expected that in a couple of years it will be possible to use it in realtime applications.

Acknowledgment

The work was supported by Russian Science Foundation grant 14-11-00308.

	SI-3		Cubic		DCCI		APLUS		SRCNN		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
14baboon	23.467	0.777	23.104	0.754	23.195	0.757	23.181	0.772	22.828	0.764	23.579	0.783
14barbara	25.895	0.828	25.929	0.829	25.309	0.817	25.387	0.806	24.758	0.792	26.014	0.835
14bridge	26.309	0.864	25.958	0.855	25.85	0.851	25.954	0.857	25.572	0.848	26.429	0.868
14coastguard	27.455	0.802	27.508	0.799	27.435	0.795	27.041	0.792	26.434	0.778	27.593	0.807
14comic	25.172	0.92	24.364	0.904	24.797	0.912	24.969	0.916	24.668	0.911	25.548	0.926
14face	31.205	0.841	31.276	0.844	31.354	0.844	30.413	0.812	29.827	0.803	31.229	0.846
14flowers	28.727	0.891	28.061	0.887	28.27	0.889	28.306	0.87	28.008	0.861	29.232	0.898
14foreman	32.567	0.895	31.623	0.897	32.428	0.901	31.43	0.857	30.958	0.853	32.959	0.906
14lenna	31.4	0.851	31.177	0.858	31.445	0.86	30.541	0.814	30.029	0.803	31.666	0.858
14man	27.559	0.864	27.049	0.858	27.305	0.861	27.205	0.848	26.702	0.84	27.656	0.86
14monarch	31.172	0.873	30.091	0.877	30.664	0.881	30.255	0.821	30.163	0.811	32.097	0.88
14pepper	31.653	0.852	31.356	0.86	31.637	0.862	30.691	0.815	30.267	0.804	31.942	0.861
14ppt3	26.877	0.916	25.223	0.899	25.449	0.903	26.375	0.874	26.973	0.882	27.629	0.927
14zebra	29.314	0.915	28.322	0.917	28.541	0.913	28.951	0.906	27.627	0.894	28.937	0.916
5baby	32.513	0.876	32.65	0.885	32.618	0.885	31.548	0.835	31.026	0.829	32.621	0.885
5bird	33.141	0.93	32.663	0.934	33.026	0.937	32.311	0.904	31.666	0.902	33.317	0.941
5butterfly	27.742	0.952	25.86	0.943	26.773	0.951	27.357	0.941	27.399	0.938	29.047	0.959
5head	31.227	0.839	31.297	0.842	31.373	0.843	30.427	0.81	29.842	0.802	31.25	0.845
5woman	30.377	0.909	29.767	0.911	30.347	0.915	29.781	0.88	29.309	0.878	30.981	0.926
Average	**29.146**	**0.873**	**28.593**	**0.871**	**28.832**	**0.872**	**28.533**	**0.849**	**28.108**	**0.841**	**29.459**	**0.880**

Fig. 1. The average PSNR and SSIM results for the test datasets for resampling with a factor of 2.

References

[1] T. Blu, P. Thévenaz, and M. Unser, "Linear interpolation revitalized," *Image Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 710–719, 2004.

[2] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.

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

[4] G. A. and A. N, "Real time artifact-free image interpolation," *IEEE Transaction on Image Processing*, vol. 20, no. 10, pp. 2760–2768, 2011.

[5] D. Zhou, X. Shen, and W. Dong, "Image zooming using directional cubic convolution interpolation," *IET Image Processing*, vol. 6, no. 6, pp. 627–634, 2012.

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

[7] H. A. Aly and E. Dubois, "Image up-sampling using total-variation regularization with a new observation model," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1647–1659, 2005.

[8] A. Lukin, A. S. Krylov, and A. Nasonov, "Image interpolation by super-resolution," in *Proceedings of GraphiCon*, vol. 2006, 2006, pp. 239–242.

[9] A. S. Krylov, A. S. Lukin, and A. V. Nasonov, "Edge-preserving nonlinear iterative image resampling method," in *Proceedings of International Conference on Image Processing (ICIP'09)*, 2009, pp. 385–388.

[10] R. Timofte, V. Smet, and L. Gool, "Anchored neighborhood regression for fast example-based super-resolution," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1920–1927, 2013.

[11] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Computer Vision–ACCV 2014*, 2014, pp. 111–126.

[12] J.-S. Choi and M. Kim, "Super-interpolation with edge-orientation based mapping kernels for low complex 2x upscaling," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 469–483, 2015.

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

[14] K. Deng, J. Tian, J. Zheng, X. Zhang, X. Dai, and M. Xu, "Retinal fundus image registration via vascular structure graph matching," *Journal of Biomedical Imaging*, vol. 2010, p. 14, 2010.

[15] J. G. Zilly, J. M. Buhmann, and D. Mahapatra, "Boosting convolutional filters with entropy sampling for optic cup and disc image segmentation from fundus images," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2015, pp. 136–143.

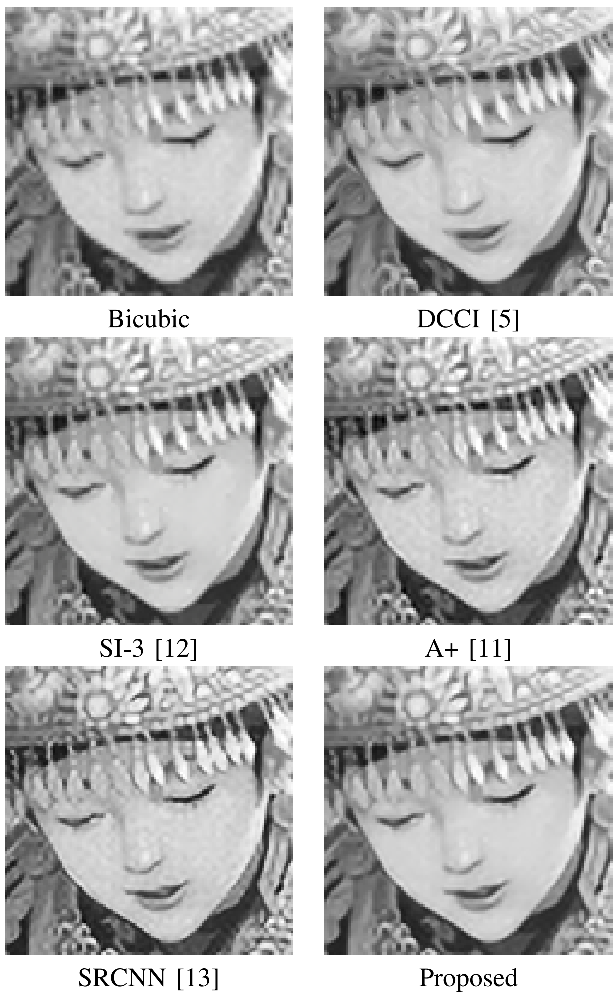

Fig. 2. Application of image upscaling for 'clown' image with noise $\sigma = 3.4$ with a factor of 2.

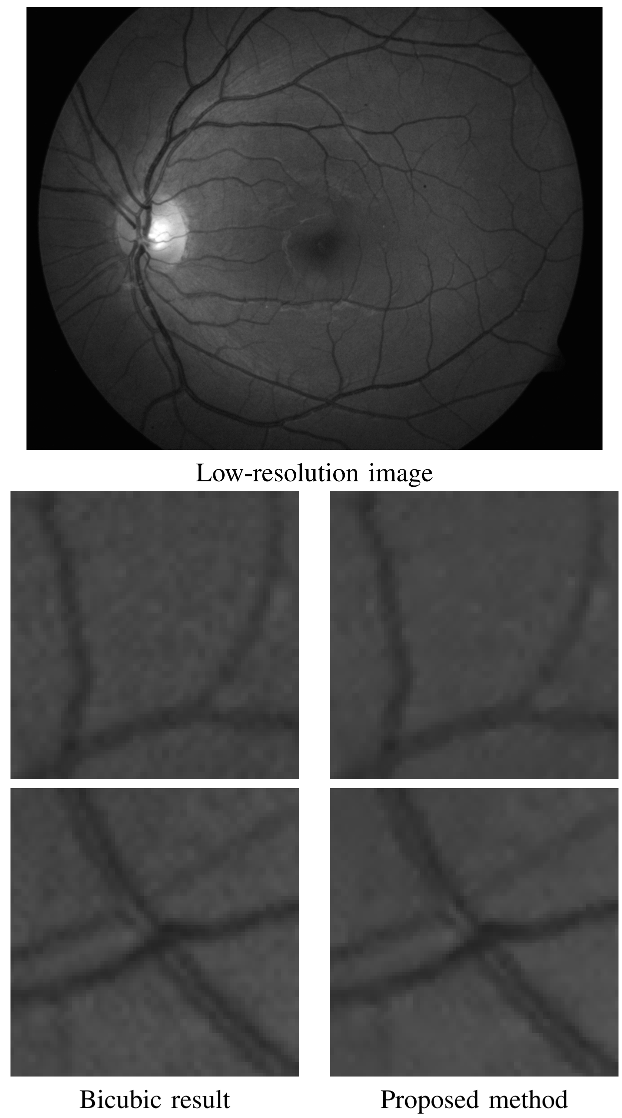

Fig. 3. The result of retinal image resampling.

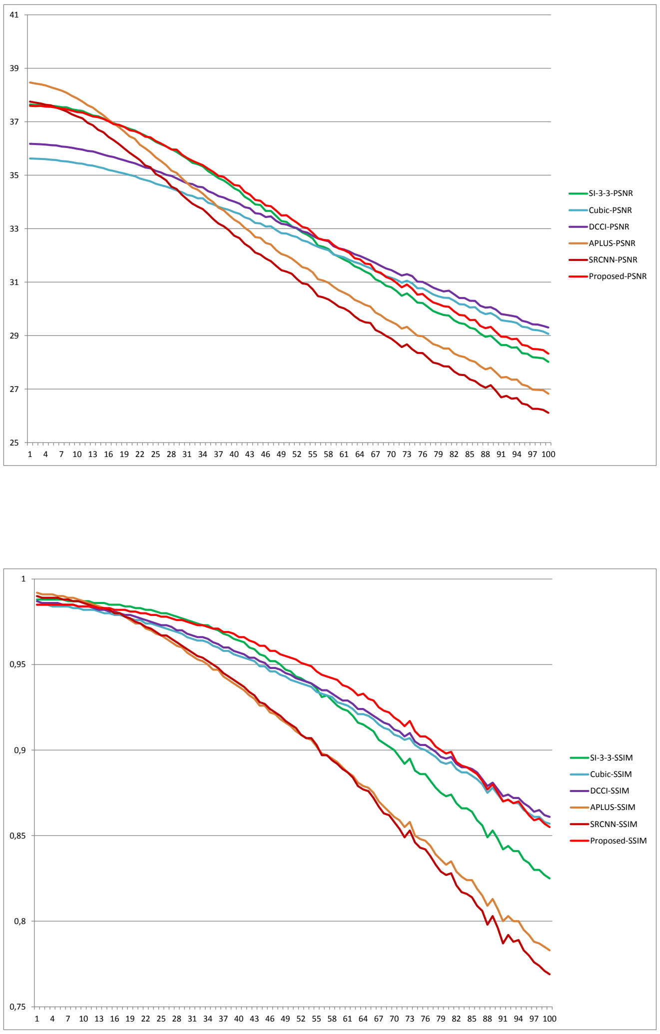

Fig. 4. The values of objective metrics against noise level for 'bird' image from Set5. The vertical axis is the metric value (PSNR or SSIM), the horizontal axis is the noise level. The value 0 corresponds to noiseless images while the value 100 corresponds to Gaussian noise with $\sigma = 10$.

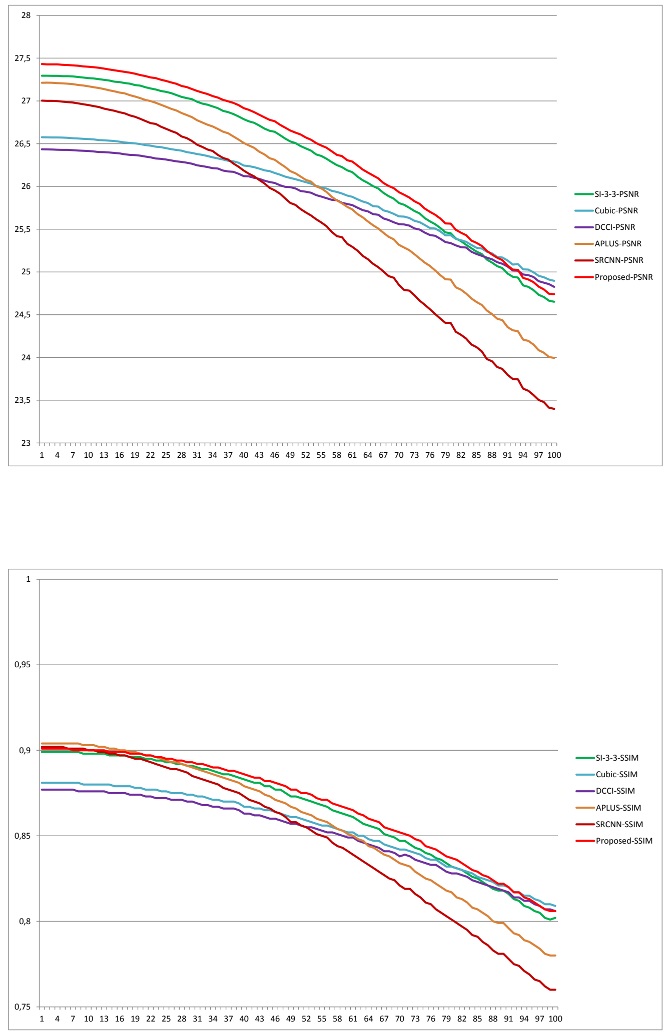

Fig. 5. The values of objective metrics against noise level for 'bridge' image from Set14.