EDGE-BASED LOSS FUNCTION FOR SINGLE IMAGE SUPER-RESOLUTION

George Seif, Dimitrios Androutsos

Ryerson University
Department of Electrical and Computer Engineering
350 Victoria St.

ABSTRACT

In recent years, convolutional neural networks have shown state-of-the-art performance on the task of single-image super-resolution. Although these proposed networks have shown high-quality reconstruction results, the use of the mean-squared error (MSE) loss function for training tends to produce images that are overly smooth and blurry. The MSE does not consider image structures that are often important for achieving high human-perceived image quality. We propose a novel edge-based loss function to improve super-resolution resconstruction of images. Our loss function directly optimizes the edge pixels of the reconstructed image, thus driving the trained network to produce high-quality salient edges and thus sharper images. Extensive quantitative and qualitative results show that our proposed loss function significantly outperforms the MSE.

***Index Terms*—** Image super-resolution, Deep neural network, convolutional neural network, edge detection

1. INTRODUCTION

Single Image Super-Resolution (SISR) is an image processing task having the aim of increasing the spatial resolution of a digital image. Prior literature generally formulates the problem as trying to reconstruct the original High Resolution (HR) image from its corresponding Lower Resolution (LR) image. The LR image lacks much of the high-frequency details from the original which represent perceptually pleasing image structures and textures. The main challenge of SISR lies in the reconstruction of these high-frequency details, given only a LR image that mainly lacks the original HR image's high-frequency components.

Learning algorithms largely dominate the state-of-the-art in SISR. Sparse coding based methods [1, 2, 3] use dictionairy learning to learn sparse signal representations for image patches. Random forests [4] and neighbour embeddings [5] based methods have also performed well on SISR.

Most prominently, deep learning based methods have been applied to SISR with great success. Dong et al. [6] proposed a Super-Resolution Convolutional Neural Network (SRCNN) that learned the direct mapping from a LR image upscaled using bicubic interpolation to the original HR image. Further improvements have been made to deep Convolutional Neural Network (CNN) architectures using residual learning [7], recursive learning [8], learned upscaling [9], or a combination of these ideas [10]. The most recent model proposed in [10] uses the Charbonnier loss functions where as the others all use the mean-squared error (MSE).

While MSE based loss functions have shown promise they do suffer from a number of drawbacks. It has been shown that MSE does not correlate well with perceptual quality as judged by human observers [11] due to the fact that the MSE does not take into account any salient features embedded within the image. Salient features such as structure and texture have been shown to be highly correlated with human-perceived image quality [12] as well as being very useful for computer vision tasks [13]. Without these structures, the network has no contextual guide to reconstruct the image, and only relies on raw pixel values.

Recently, a few alternative loss functions have been proposed to address the drawbacks of the MSE by leveraging different kinds of salient features. Johnson et al. [14] proposed a feature-based loss function where the loss is the Euclidean distance between feature representations of the orignal HR image and the LR reconstructed image. Their loss function proved to be qualitatively effective at reconstructing perceptually important image features but did not perform well quantitatively in terms of PSNR and SSIM due to it's main focus being on the features and not at all on direct pixel values. Frosio et al. [11] presented a study on training CNNs for general image restoration using various loss functions including MSE, mean-absolute error (MAE), and MS-SSIM (multiscale SSIM). They found that a combination of a pixel and structural loss i.e a weighted sum of MAE and MS-SSIM was effective in increasing the quantitative PSNR/SSIM metrics as well as maintaining perceptual quality. Our edge-based loss function is inspired by [11] but is more intuitive than the MS-SSIM and allows us to train the network to reconstruct the exact image structures that are desired. Lin et al. [15] proposed to train a network to output two images: a binary edge map and the reconstructed image where the total loss is the MSE of both of these images. There are a couple of drawbacks to this approach. Firstly, the edge map is produced separately

from the reconstructed image and thus the network has to balance optimization for two totally different outputs; because the edge map is being predicted seperately from the final reconstructed image, the edges of the reconstructed output are not being directly optimized. Secondly, the predicted edge map uses ground truth binary edges and thus when computing its loss function the exact pixel values of those edges are not being taken into account.

We propose an edge-based loss function for SISR to address the drawbacks with MSE and to improve upon previous attempts at designing a loss function that is in coherence with image structures and contrast. Our loss function optmizes the deep network parameters such that image edges are given more significant importance. We achieve this by learning the exact edge pixel values directly from the reconstructed image. This ensures that the network reconstructs high-quality and accurate edges that are most important in improving perceptual image quality [12], as well as aiding in vision tasks [13]. Our loss function can be used to train any CNN and is thus versatile to be applied to any fully-convolutional image restoration task. Extensive quantitative and qualitative experiments show that our loss function significantly improves SISR reconstruction over the MSE on the same CNN architecture.

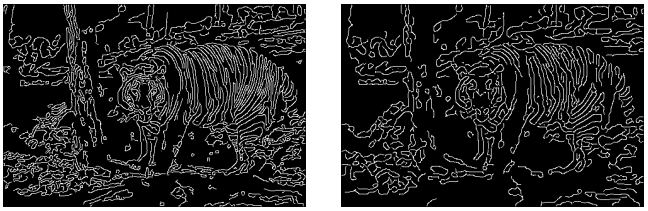
2. EDGE-BASED LOSS FUNCTION

2.1. Methodology

Given an LR image that is upscaled using bicubic interpolation, we wish to design a loss function that can be used to train a deep CNN to reconstruct the corresponding HR image. Our loss function is inspired by previous ideas of combining a pixel-based loss with a structural loss [11, 15]. The pixel-based loss component promotes overall accuracy of the reconstructed image i.e for pixel values to be directly similar to those of the original. This insures that there are no major changes in colour, lighting, or overall contrast of the image. The structural loss component guides the network to produce salient image structures that would be considered perceptually important to human observers. In particular, we propose to use edges as the basis for our structural loss.

The main challenge in SISR is the reconstruction of salient edges. Consider the edge maps shown in Figure 1. The original HR edge map shows many salient edges where as in the LR X4 scale reconstructed image, many of those edges are not present. Due to the fact that purely pixel-based loss functions equally weight all image pixels in the loss function, the network is not being optimized in a way that directly promotes the reconstruction of these salient edges.

Since salient edges are important in terms of perceptual image quality, we propose that edges should be given extra weight in the loss function via an edge-loss component. Consider a training example where X is the bicubic upscaled LR image with width W and height H, Y is the original HR image, and E is the corresponding HR edge map. To form the edge-loss component, we first apply a Canny edge detector (using Matlab default parameters) to the original HR training image Y to get the edge map E. The Matlab default Canny edge detector computes the high edge threshold by computing the gradient map of the image, followed by constructing a normalized histogram of the edge gradients, and then determining which high threshold would make at lease 70 percent of the pixels non-edge pixels; the low threshold is selected as 0.4 times the high threshold. The edge loss component is then computed as the mean of the product of the binary edge map and the reconstruction error:



(a) HR Edge Map (b) LR x4 Edge Map

Fig. 1: Edge maps of an HR image and its corresponding reconstructed LR image at x4 scale, obtained using the default Matlab Canny edge detector.

$$loss_{edges} = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H} E_{i,j} \cdot (|Y_{i,j} - X_{i,j}|)}{WH}. \quad (1)$$

To obtain the final loss, the edge loss component is summed with a pixel loss component that is the mean-abosulte error (MAE) between the original HR image and the reconstructed LR image:

$$loss_{pixels} = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H} (|Y_{i,j} - X_{i,j}|)}{WH} \quad (2)$$

$$loss_{total} = \alpha \cdot loss_{pixels} + (1 - \alpha) \cdot loss_{edges} \quad (3)$$

where α is used to control the weightg of each loss component, and we empirically set $\alpha = 0.7$. The edge component drives the network to reconstruct edges that are closer to those of the original, thus giving the network a structural edge-guidance. The pixel loss maintains the lighting and contrast of the original HR image. We use the MAE rather than MSE because with its extra squaring, the MSE heavily penalizes larger errors while having less of an effect on smaller errors, regardless of the image features or structures. Using MSE for the edge loss would over-emphasize the edge errors causing perceptually important aspects of the image such as overall lighting or contrast to have less overall weight in the total loss. Thus MAE allows for balanced weighting of the two loss components. Lai et al. [10] and Zhao et al. [11] also

report more successful results when training with an MAE loss variant over MSE.

By computing the edge loss component from the reconstructed LR image rather than as a seperate output [15], the final reconstructed image is being directly optimized to have more similar salient edges as the original HR image. Moreover, as a loss function it can be used to train any image restoration network to produce more salient edges. Critically, our edge loss component is computed using the exact pixel values of the edges, rather than a binary edge map. This ensures that edges similar to the original are being reconstructed. When a binary edge map is used [15] the edge pixel values are not taken into account, which can produce edges that are perceptually different from those of the HR image. Furthermore edge loss function is intuitively simple to understand and can further be improved using ground truth edges rather than a Canny edge detector.

2.2. Implementation and training details

As a baseline model we use the state-of-the-art VDSR architecture trained with the MSE [7]. We use the same training data as VDSR using 91 images from Yang et al. [2] and 200 images from the Berkeley Segmentation Dataset [16]. The training data is augmented in the same way as original VDSR using rotations and flips and uses the same patch size of 41x41 pixels. We train our network using the Adam optimizer [17] with learning rate 10^{-4}. The implementation is done in Keras using Tensorflow backend. PSNR and SSIM are computed by first converting colour images to YCbCr colour space and evaluating only the luminance channel for fair comparison against other methods. The final colour images are obtained by applying bicubic interpolation to the Cb and Cr channels, concatenating with the super-resolved luminance channel, and converting to RGB format.

3. EXPERIMENTAL RESULTS

We evaluate our loss function on well-known SISR benchmark datasets: Set5 [18], Set14 [19], BSDS100 [20], and Urban100 [21]. We conduct and present both quantitative and qualitative evaluations. As shown in Table 1, our loss function with VDSR outperforms the original VDSR trained with MSE. Furthermore, training VDSR with our edge-based loss function allows it to outperform other architectures that would normally perform better with their selected loss functions, such as DRCN [8] with MSE and LapSRN [10] with Charbonnier. It can also be observed that our edge-loss achieves larger improvements over MSE with images that have many edges. In particular, our loss shows the largest improves on a X2 scale because the input image, a X2 bicubic upscale LR image, has many salient edges that are already close to the original HR ones. For larger scales such as X4 many of the original HR edges are either heavily distorted or completely gone from the image, thus making edge reconstruction more challenging.

We also conducted a survey as a qualitative evaluation of our loss function. We first super-resolved every image in the Urban100 dataset at an X3 scale using two networks trained separately with our loss and MSE respectively. We cropped the center 200x200 pixel patch of each image such that the difference in restoration performance can clearly be seen. Using Amazon Mechanical Turk (AMT) we conducted our survey by asking users to select the restored image that is the clearest and sharpest. We collected a total of 8000 user submissions from the Urban100 dataset and in Figure 2 plot a histogram of the proportion of participants preferring our loss function over MSE for each of the 100 image pairs. From the histogram, for 65 out of the 100 images in the survery, at least 50% of participants selected the image restored using our loss function as clearer and sharper. Additionally, many images had a strong majority vote with at least 60% and 70% of participants selecting the image restored using our loss function as clearer and sharper for 44 and 27 of the 100 test images, respectively. We also computed the average participant selection for all 100 images and found that overall 63% of partcipants selected the image produced using our loss function as being clearer and sharper.

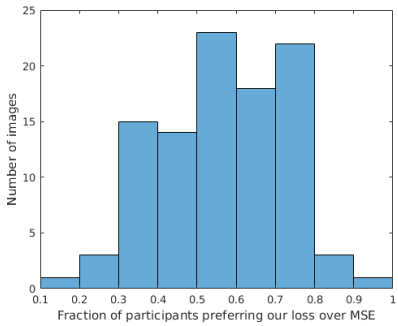


Fig. 2: Histogram of the proporations of participants preferring our loss function over MSE for each of the 100 image pairs in the Urban100 dataset

Figures 3, 4, and 5 also show some upscaling results for X2, X3, and X4 scales, respectively. Here we compare the performance of MSE vs. our edge loss directly by showing the results from training VDSR. This provides the most direct comparison between the two to evaluate if our loss function indeed outperforms MSE using the same network. As can be seen, our loss function optimizes the network to properly reconstruct salient edges, without sacrificing anything in lighting or contrast. The edge guidance provides the most improvement when edges in the original HR image are most clearly defined. Our results can also further be improved by using ground-truth, human labelled edges rather than the Canny edge detector.

Table 1: Quantitative evaluation of state-of-the-art SR algorithms using **PSNR/SSIM**

Dataset	*Scale*	*Bicubic*	*SRCNN*	*VDSR*	*DRCN*	*LapSRN*	*Ours*
Set5	2	33.66/0.9299	36.66/0.9542	37.53/0.9587	37.63/0.9588	37.52/0.959	**37.70/0.9602**
	3	30.39/0.8682	32.75/0.9090	33.66/0.9213	33.82/0.9226	33.82/0.922	**33.82/0.9228**
	4	28.42/0.8104	30.48/0.8628	31.35/0.8838	31.53/0.8854	**31.54/0.885**	31.45/0.8848
Set14	2	30.24/0.8688	32.42/0.9063	33.03/0.9124	33.04/0.9118	33.08/0.913	**33.28/0.9142**
	3	27.55/0.7742	29.28/0.8209	29.77/0.8314	29.76/0.8311	29.87/0.832	**29.93/0.8322**
	4	26.00/0.7027	27.49/0.7503	28.01/0.7674	28.02/0.7670	**28.19/0.772**	28.15/0.7708
BSDS100	2	29.56/0.8431	31.36/0.8879	31.90/0.8960	31.85/0.8942	31.80/0.895	**31.98/0.8961**
	3	27.21/9.7385	28.41/0.7863	28.82/0.7976	28.80/0.7963	28.82/0.798	**28.88/0.7985**
	4	25.96/0.6675	26.90/0.7101/	27.29/0.7251	27.23/0.7233	27.32/0.728	**27.35/0.7254**
Urban100	2	26.88/0.8403	29.50/0.8946	30.76/0.9140	30.75/0.9133	30.41/0.910	**30.94/0.9153**
	3	24.46/0.7349	26.24/0.7989	27.14/0.8279	27.15/0.8276	27.07/0.828	**27.28/0.8288**
	4	23.14/0.6577	24.52/0.7221	25.18/0.7524	25.14/0.7510	25.21/0.756	**25.28/0.7568**

(a) Original HR image

(b) Ours (w/ edge loss)

(c) VDSR (w/ MSE)

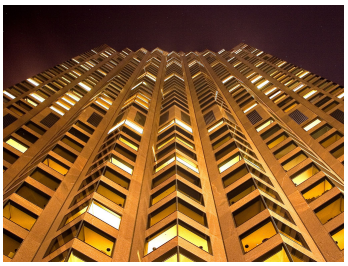
Fig. 3: X2 SR results on the 'monarch' image from Set14

(a) Original HR image

(b) Ours (w/ edge loss)

(c) VDSR (w/ MSE)

Fig. 4: X3 SR results on the '108005' image from BSDS100



(a) Original HR image

(b) Ours (w/ edge loss)

(c) VDSR (w/ MSE)

Fig. 5: X4 SR results on the '$\text{img}_0 15'$ $image from BSDS100$

4. CONCLUSION

We have presented an edge-based loss function to address the challenge of reconstructing salient edges in SISR. In contrast to other approaches, our method uses well defined edges as a structural guide for network training to aid in optimal edge reconstruction. Experimental results have shown that our loss function outperforms others used in state-of-the-art models both quantitatively and qualitatvely. Our results can further be improved by using ground-truth edges.

5. REFERENCES

[1] Radu Timofte, Vincent De Smet, and Luc Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[2] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[3] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma, "Image super-resolution as sparse representation of raw image patches," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[4] Samuel Schulter, Christian Leistner, and Horst Bischof, "Fast and accurate image upscaling with super-resolution forests," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[5] Hong Chang, Dit-Yan Yeung, and Yimin Xiong, "Super-resolution through neighbor embedding," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, vol. 1, pp. I–I.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[7] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] Chao Dong, Chen Change Loy, and Xiaoou Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision*. Springer, 2016, pp. 391–407.

[10] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[11] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[12] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[13] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[15] Yukai Shi, Keze Wang, Li Xu, and Liang Lin, "Local- and holistic-structure preserving image super resolution via deep joint component learning," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[16] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. IEEE, 2001, vol. 2, pp. 416–423.

[17] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[19] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[20] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[21] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, "Single image super-resolution from transformed self-exemplars," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.