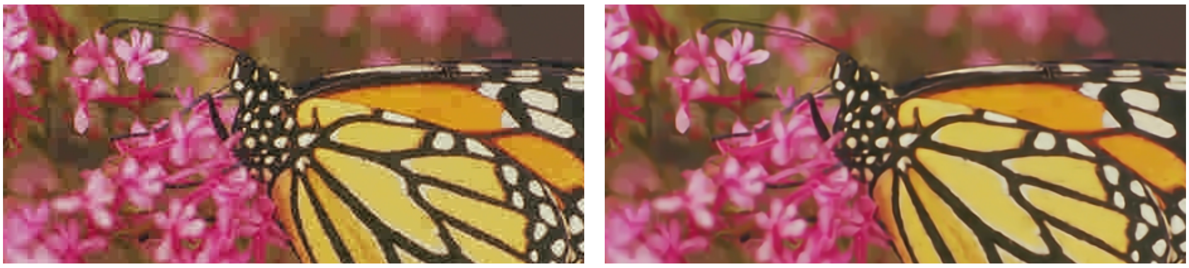
Compression Artifacts Removal Using Convolutional Neural Networks

Pavel Svoboda Michal Hradis David Barina Pavel Zemcik

Faculty of Information Technology
Brno University of Technology
Bozetechova 1/2, Brno
Czech Republic
{isvoboda,ihradis,ibarina,zemcik}@fit.vutbr.cz



ABSTRACT

This paper shows that it is possible to train large and deep convolutional neural networks (CNN) for JPEG compression artifacts reduction, and that such networks can provide significantly better reconstruction quality compared to previously used smaller networks as well as to any other state-of-the-art methods. We were able to train networks with 8 layers in a single step and in relatively short time by combining residual learning, skip architecture, and symmetric weight initialization. We provide further insights into convolution networks for JPEG artifact reduction by evaluating three different objectives, generalization with respect to training dataset size, and generalization with respect to JPEG quality level.

Keywords
Deep learning, Convolutional neural networks, JPEG, Compression artifacts, Deblocking, Deringing

1 INTRODUCTION

This work presents a novel method of image restoration using convolutional networks that represents a significant advancement compared to the state-of-the-art methods. We study the direct approach [12] in which a fully convolutional network accepts a degraded image as input and outputs a high quality image. By making a number of important improvements regarding the network architecture, initialization, and training, we are able to train large and deep networks for JPEG compression artifact reduction which surpass the state-of-the-art in this task. The networks predict a residual image [16] describing changes to be applied to the input image, and they incorporate skip connections [18] which allow information to bypass the middle layers. We reduce the "saturation" of ReLU units in deeper layers by centering filters during network initialization which allows us to use significantly faster learning rates.

Lossy image compression achieves high compression ratios through elimination of information that does not contribute to human perception of images, or contributes as little as possible. Due to the limitations of the human visual system, such loss of information may be acceptable in many scenarios but the introduced visual artifacts become unacceptable at higher compression ratios. The primary methods currently used for lossy image compression include JPEG and JPEG 2000. This paper focuses on the JPEG compression method [13] and the degradation it causes. The JPEG compression chain consists of a block-based discrete cosine transform (DCT), followed by a quantization step utilizing a quantization matrix, and an entropy coding. The decompression follows this process in reverse order.

Blocking, blurring, and ringing artifacts are typical examples of image degradation caused by the lossy compression methods. Considering the JPEG method, the degradation is the result of information loss in the DCT coefficient quantization step. More specifically, the blocking artifacts are caused by the grid segmentation into 8×8 cells employed in the JPEG standard and the resulting discontinuities at the cell edges. The ringing artifacts (or the Gibbs phenomenon) are induced oscillations caused by removal of high frequencies during the quantization. The removal of high frequencies causes blurring as well, but the blurring is less noticeable compared to the ringing artifacts. Blocking is mostly noticeable in low-frequency regions, while the ringing artifacts are especially well noticeable around sharp edges.

The convolutional networks have to learn to recognize the compression artifacts and fill them appropriately with respect to the neighboring image content. In this sense, the networks incorporate both the data term and prior regularization term of standard image restoration techniques, and they can make use of correlations between image content and the image degradation.

Convolutional networks have been successfully used in many image restoration tasks including super resolution [4, 16], denoising [15], structured noise removal [6], non-blind deconvolution [21, 30], blind deconvolution in specific image domains [12, 24], and sub-tasks of blind deconvolution [20]. Our work was mostly inspired by the large deblurring networks of Hradis *et al.* [12], and by Kim *et al.* [16] who showed that residual learning together with good weight initialization enabled training of large convolutional networks for super resolution. We extend the work of Dong *et al.* [4] who achieved state-of-the-art compression artifact reduction even with very small convolutional networks. However, they were not able to scale up their networks due to problems with training convergence.

2 RELATED WORK

A large number of methods designed to reduce compression artifacts exist ranging from relatively simple and fast hand-designed filters to fully probabilistic image restoration methods with complex priors [29] and methods which rely on advanced machine learning approaches [4].

Simple deblocking and artifact removal postprocessing filters are included in most image and video viewing software. For example, the FFmpeg framework includes the simple postprocessing (spp) filter [19] which simply re-applies JPEG compression to the shifted versions of the already-compressed image, and averages the results. The spp filter uses the quantization matrix (compression quality) of the original compressed image as the matrix has to be stored with the image to allow for decompression. Pointwise Shape-Adaptive DCT (SA-DCT) [7, 8], in which the thresholded or attenuated transform coefficients are used to reconstruct a local estimate of the signal within the adaptive-shape support, is currently considered the state-of-the-art deblocking method. However, similarly to other deblocking methods, SA-DCT overly smooths images and it is not bale to sharpen edges. In video compression domain, advanced in-loop filters (deblocking and SAO filters) known from video compression standards like H.264 or H.265 are obligatorily applied. A completely different deblocking approach was presented in [31], where the authors applied DCT-based lapped transform on the signal already in the DCT domain in order to undo the harm done by the DCT domain processing. However, the video in-loop deblocking methods, SA-DCT deblocking (only to estimate parameters), and methods derived from the lapped DCT rely on the cognizance of the DCT grid. Unlike these methods, the method proposed in this paper is able to process images without such knowledge.

This work focuses on application of convolutional networks to reconstruction of images corrupted by JPEG compression artifacts. Convolutional networks belong to an extensively studied domain of deep learning [2]. Recent results in several machine learning tasks show that deep architectures are able to learn the high level abstractions necessary for a wide range of vision tasks including face recognition [25], object detection [9], scene classification [17], pose estimation [26], image captioning [27], and various image restoration tasks [4, 16, 15, 6, 21, 30, 12, 24, 20]. Today, convolutional networks based approaches show the state-of-the-art results in many computer vision fields.

Small networks were historically used in image denoising and other tasks. On the other hand, deep and large fully convolutional networks have become only recently important in this field. Burger *et al.* [3] used feed forward three layer neural network for image denoising. While there were attempts to use neural networks for denoising before, Burger *et al.* showed that this approach can produce state-of-the-art results when trained on a sufficiently large dataset.

A non-blind deconvolution method of Schuler *et al.* [21] uses a regularized inversion of the blur kernel in Fourier domain followed by a multi-layer perceptron (MLP) based denoising step. The shortcoming of the approach is that a separate MLP models have to be trained for different blur kernels, as a general models trained for multiple blur kernels provide inferior reconstruction quality. Schuler *et al.* [20] introduced a learning based approach to blind deconvolution. They perform a regression from the blurred image towards the source blur kernel. The neural network itself is trained to extract image features useful for estimation of the blur point spread function. Sun *et al.* [23] presented CNN-based approach for non-uniform motion blur removal which classified image patches into closed set of blur kernel types. The local classification outputs were used as input to a Markov random field model which estimates the dense non-uniform motion blur field over the whole image. Hradis *et al.* [12] trained CNNs composed of only convolutional layers and rectified linear units (ReLU) to directly map blurred and noisy images of text images to high quality clean images. The approach was extended by Svoboda *et al.* [24] who demonstrated high quality deblurring reconstructions for car license plates in a real-life traffic surveillance system. Their results show that a single CNN can be trained for a full range of motion blurs expected to appear in a specific traffic surveillance camera resulting in a robust and fast system.

Dong *et al.* [4] introduced super-resolution convolutional neural network (SRCNN) to deal with the ill-posed problem of super-resolution. The SRCNN is designed according the classical sparse coding methods – the three layers of SRCNN consist of feature extraction layer, a high dimensional mapping layer, and a final reconstruction layer. The very deep CNN based super-resolution method proposed by Kim *et al.* [16] builds on the work of Dong *et al.* [4] and it shows that deep networks for super-resolution can be trained when proper guidelines are followed. They initialized networks properly and they used so-called residual learning in which the network predicts how the input image should be changed instead of predicting the desired image directly. Residual learning appears to be very important in super-resolution. The resulting 20 layers deep networks trained with adjustable gradient clipping significantly outperform previous approaches. However, it is unclear how effective residual learning would be in other image processing tasks where the networks inputs and outputs are not correlated that strongly as in super-resolution. We follow this approach in our work on JPEG reconstruction.

Convolutional networks have previously been used for suppressing compression artifacts by Dong *et al.* [5], who proposed a compact and efficient CNN based on SRCNN – artifacts removing convolutional network (AR-CNN). AR-CNN extends the original architecture of SRCNN with feature enhancement layers. The network training consist of two stages – a shallow network is trained first and it is used as an initialization for a final 4 layer CNN. As reported in the paper, this two stage approach improved results due to training difficulties encountered when training the full 4 layer network from scratch. The authors also state that they aim to achieve feature enhancement instead of just making the CNN deeper. They argue that although the deeper SRCNN introduces a better regressor between the low-level features and the reconstruction, the bottleneck lies on the features. Thus the extracting layer is augmented by the enhancement layer which together may provide better feature extractor.

We adapt the idea of residual learning [16] for the JPEG compression artifact removal based on CNN. We follow the assumption "deeper is better" and we try to learn our deep residual CNNs in a single step by creating a new recipe including initialization, network architecture, and high learning rates. The resulting networks significantly outperform the classical JPEG compression artifact removal methods, as well as, the AR-CNN [5] on common dataset measured by PSNR, specialized deblocking assessment measure PSNR-B, and SSIM.

3 CNN IMAGE ENHANCEMENT

In computer vision, CNNs are most extensively studied in the context of classification, semantic class segmentation, object detection, and captioning where the networks are often constrained to a fixed input size. This is due to the fully connected layers which are used as the final layers in order to aggregate information from a whole image. In low level image processing (but not limited to it), the so-called fully convolutional neural networks [18] (FCN) are preferred as they behave as non-linear convolutional operators – they process each image position the same way and they can be applied to images of arbitrary size.[1] The architecture of fully convolutional networks is limited to convolutional operations (linear convolution, so-called deconvolution, local response normalization, and local pooling) and element-wise operations. Most image processing networks use only convolutions and element-wise non-linearities (ReLU, sigmoid, tanh) [12, 24, 5, 16, 4, 21, 20]. In the case that no pooling and no deconvolution layer is used, the size of the input is reduced only by size of the convolution layer kernels (by the size of receptive field).

The fully convolutional networks F used in our work consist of an input data layer F_0, convolutional layers F_ℓ, where $0 < \ell \leq L$ with F_ℓ weights represented as convolutional kernels W_ℓ with their biases b_ℓ, and element-wise max operations (ReLU) as follows:

$$\begin{aligned} F_0(y) &= y \\ F_\ell(y) &= \max(0, W_\ell * F_{\ell-1}(y) + b_\ell) \\ F(y) &= W_L * F_{L-1}(y) + b_L \end{aligned} \tag{1}$$

Where y is the distorted input image and $F(y)$ is the restored output image.

[1] In practice, the minimum size of processed images is constrained by the receptive field size of the network.

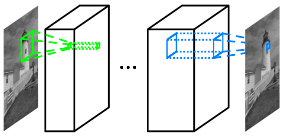


Figure 1: Illustration of a network with direct architecture.

We use the standard mean squared error (MSE) objective function

$$\frac{1}{n}\sum_{i=1}^{n}\|F(y_i)-x_i\|_2^2, \tag{2}$$

which is often used for general image enhancement. It is computed on a training data represented as pairs (y_i, x_i), $0 < i \leq n$, where y_i represents the reconstructed image and x_i its corresponding clean image.

Direct mapping objective. In direct mapping shown in Figure 1, the networks learn to transform corrupted images directly to clean images. This approach leads to high quality results in specific low level image processing tasks i.e. in blind and non-blind deconvolution for text denoising or motion deblurring [12, 24], in super-resolution [4] or JPEG compression artifacts reduction [5]. Direct mapping forces the network to transfer the whole image through all its layers until it reaches the output. The learning of such autoencoder-like mapping in situations where the input images are highly correlated with the desired outputs may be wasteful especially for large and deep networks. It may be one of the main reasons why Dong *et al.* [5] were not able to scale up their networks and why they required approximately 10^7 iterations to train their AR-CNN. Similar problems were reported by Kim *et al.* [16].

Residual objective. The residual objective was originally introduced for super-resolution [16] where the input and output images are highly correlated. Instead of learning to predict the output image, the network in residual learning learns the changes which should be applied to the input image – it predicts the residual image $r = y - \hat{x}$ between the distorted y and latent high-quality image $\hat{x}$. The residual learning scheme is depicted in Figure 2. Kim *et al.* [16] were able to speed up the training by large factor of (up to $10^4\times$) with residual learning and it allowed them to learn much deeper networks – 20 layers vs. 3 in [4] and 4 in [5].

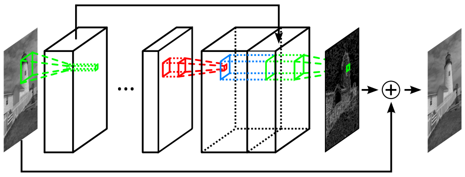


Figure 2: Illustration of a network with skip architecture and residual loss.

Edge emphasized objective. Mean square error used in many image restoration methods does not necessarily correlate well with the image quality perceived by humans. With convolutional networks, it is relatively easy to use more perceptually valid error measures as objective functions, as long as they can be efficiently differentiated (e.g. SSIM). We decided to add partial first derivatives of the image to the loss function in a form of vertical and horizontal Sobel filters. This is achieved by adding the objective function computed on image derivative calculated by Sobel filter G as

$$\frac{1}{n}\sum_{i=1}^{n}\|G*F(y_i)-G*x_i\|_2^2. \tag{3}$$

Our assumption is that the addition of the first derivatives should force the network to focus specifically on high frequency structures such as edges, ringing artifacts, and block artifacts and it could lead to perceptually better reconstructions. The combined edge emphasized loss can be easily implemented in all existing convolutional network frameworks by defining the derivative Sobel filters as a convolutional layer with predefined fixed filters. The network utilizing such objective function is shown in Figure 3.

Symmetric weight initialization. Weights in convolutional networks are usually initialized by sampling from some simple distribution (e.g. Gaussian or uniform) with mean equal to 0. The zero mean is desirable as it prevents mean offsets of activations to propagate through the layers. In case the mean was not zero, any mean offset in input values would result in non-zero mean of output activations which could force the ReLU non-linearities to get fully stuck either in the positive linear interval or, even worse, in the negative interval where gradients are not propagated rendering the unit useless.

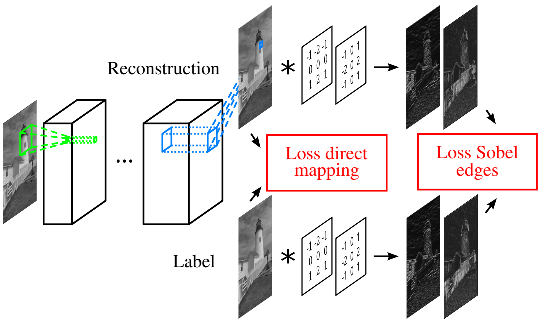


Figure 3: Illustration of a network with edge preserving loss.

Layer	1	2	3	4(+1)	5	6(+1)	7	8
Filter size	11×11	3×3	3×3	3×3	1×1	5×5	1×1	5×5
Channels	32	64	64	64(+32)	64	64(+32)	128	1

Table 1: L8 architecture – filter size and number of channels for each layer.

Although the weights are sampled from a distribution with zero mean, the means of individual convolutional filters are not zero due to the fact that they are a finite sample from the distribution. These random offsets together with the positive offset of ReLU activations cause units in deeper layers to become more likely to be either permanently turned off or turned on, which increases sparsity of the activations and increases effective mean offsets of the deeper layers. The result is that that majority of units in deep layers become almost useless right after the initialization.

Some activation normalization methods, such as "batch normalization" [14], can eliminate the saturation problem, but the normalization introduces noise during training which is not desirable for image restoration networks.

We eliminate this problem by explicitly forcing individual filters to have zero mean during initialization. Such initialization allows us to use significantly higher initial learning rates, especially together with residual learning, and it results in trained networks with significantly fewer saturated neurons.

We could explicitly force all filters to have zero mean during the whole training. Such constraint almost entirely eliminates any potential for unit saturation, but it prevents networks to utilize the DC component of input signals. Although we were able to achieve reasonably good results with this constraint in our preliminary experiments, we did not find it necessary and it was not used in the experiments presented in this paper.

Skip architecture. Deeper networks may have problems with exploding and vanishing gradients and they may take a long time to learn to efficiently propagate information through large number of layers. The problems with the gradients can be eliminated by proper initialization [10]. The problems with propagating information through many layers can be alleviated by bypassing some layers [18] or by letting layers to learn residual of their inputs [11]. The skip architecture with the residual objective function is shown in Figure 2.

We employ a skip architecture similarly to Long *et al.* [18]. We feed activations of the first convolutional layer to some deeper layers bypassing the layers inbetween. Unlike Long *et al.* [18] who add the activations together, we concatenate them. The goal of the skip architecture is to allow the network to pass geometric information easily from the input to the output, and to allow for more complex reasoning about the image content in the middle layers (e.g. what is an artifact and what local context information should be used to repair the artifacts.

Layer	1	2	3	4
Filter size	11×11	3×3	3×3	5×5
Channels	48	64	64	1

Table 2: L4 architecture – filter size and number of channels for each layer.

Network architectures. This paper presents two different FCN architectures which use only convolutional units and ReLU non-linearities. The first architecture denoted as L4 is relatively small with four layers defined in Table 2. The second network, denoted as L8, has eight layers and it utilizes the skip architecture by concatenating activation of the first layer with activations of the fourth and sixth layers. The exact definition of L8 is in Table 1. The receptive fields of L4 and L8 are 19×19 and 25×25, respectively.

4 EXPERIMENTAL RESULTS

All the experiments were computed on images from BSDS500 [1] and LIVE1 [22] datasets. The networks were trained solely on the merged train and test part of BSDS500 which contain 400 images. The images were transformed to gray-scale using the YCbCr color model by keeping the luma component – Y only. Althought the networks can process color images, we evaluate on gray-scale images because we focus on the ringing and blocking artifacts and not on the chromatic distortions. The gray-scale images were compressed with the MATLAB JPEG encoder into six disjoint sets according the JPEG quality. Specifically, we use images compressed with the quality 10, 20, 30, 40, 50, and 60.

The networks were evaluated on the validation set from BSDS500 which includes 100 high quality compressed images and on the LIVE1 dataset containing 29 color images (uncompressed BMP format). All the evaluation images were transformed to gray-scale the same way as the training images and compressed using the same encoder.

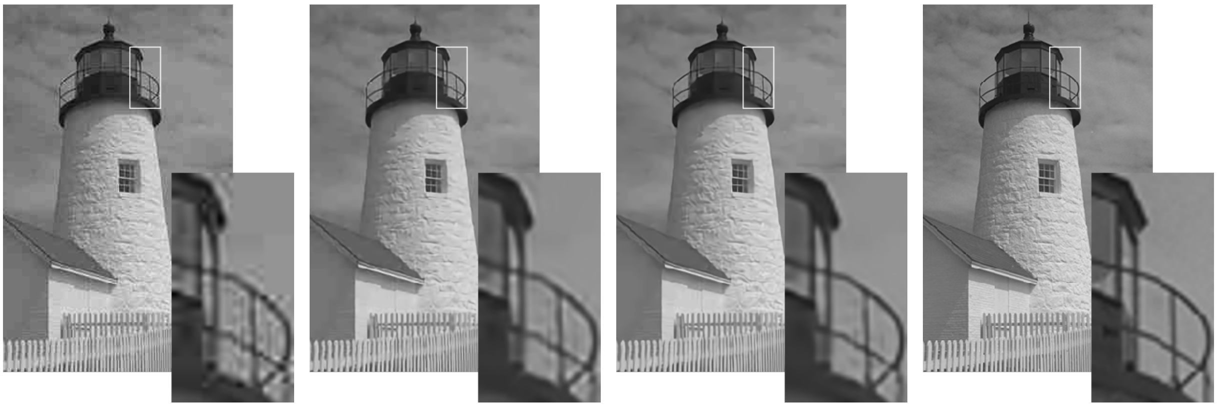
Several metrics for objectively assessing perceptual quality of images exist. We use PSNR, PSNR-B, and SSIM. Generally, the most commonly used quality metric is the mean squared error (MSE). This quantity is computed by averaging squared intensity differences

method	Q10			Q20		
	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
distorted	27.77	25.33	0.791	30.07	27.57	0.868
spp	28.37	27.77	0.806	30.49	29.22	0.877
SA-DCT	28.65	28.01	0.809	30.81	29.82	0.878
AR-CNN	28.98	28.70	0.822	31.29	30.76	0.887
L4 Residual	**29.08**	**28.71**	**0.824**	31.42	30.83	0.890
L8 Residual	–	–	–	**31.51**	**30.92**	**0.891**

Table 3: Image reconstruction quality on LIVE1 validation dataset for JPEG quality 10 and 20.

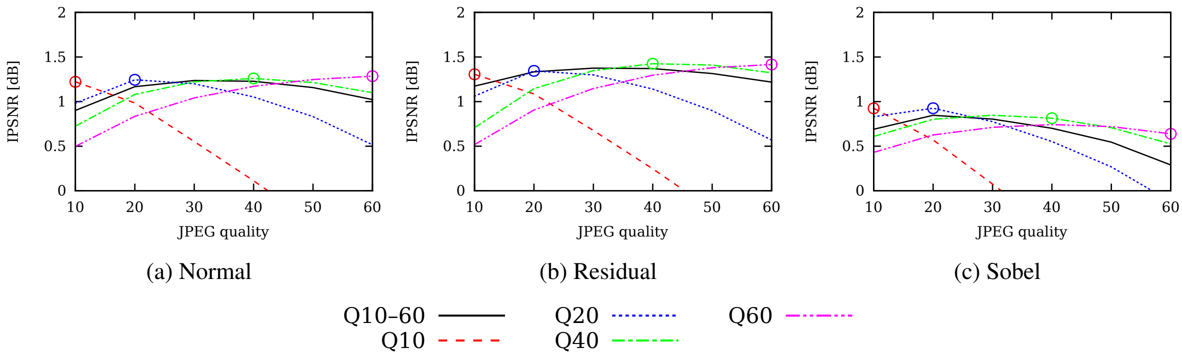
method	Q10			Q20		
	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
distorted	27.58	24.97	0.769	29.72	26.97	0.852
spp	28.13	27.49	0.782	30.11	28.68	0.859
AR-CNN	28.74	**28.38**	0.796	30.80	30.08	0.868
L4 Residual	**28.75**	28.29	**0.800**	30.90	30.13	0.871
L8 Residual	–	–	–	**30.99**	**30.19**	**0.872**

Table 4: Image reconstruction quality on BSDS500 validation dataset for JPEG quality 10 and 20.



(a) distorted (b) AR-CNN (c) L08 (d) original

Figure 4: Illustrative comparison of reconstruction quality on lighthouse3 image from LIVE1 dataset, JPEG quality 20.



(a) Normal (b) Residual (c) Sobel

Figure 5: Generalization ability of L4 networks trained with Normal, Residual, and Edge preserving objectives for different JPEG quality levels.

of the distorted image and the reference image. The quantity is often expressed in a logarithmic scale as the peak signal-to-noise ratio (PSNR). Unfortunately, PSNR and MSE are not necessarily correlated well with perceptual quality. The structural similarity index (SSIM) [28] that compares local patterns of pixel intensities should be better correlated with perceptual quality. Since we focus on JPEG artifacts which include blocking artifacts, a block-sensitive metric referred to as the PSNR-B [32] should provide additional insights. PSNR-B modifies the original PSNR by including an additional blocking effect factor (BEF). Some experiments report IPSNR which is a PSNR increase compared to PSNR of the degraded image. IPSNR is more stable across different dataset and it directly reflects the quality improvement.

We compare our results to AR-CNN [5], to the widely regarded deblocking oriented SA-DCT [7, 8], and to a simple postprocessing filter spp included in the FFmpeg framework [19]. While L4 was used in most experiments and it was trained for various compression quality levels, L8 was trained only for quality 20. If not stated otherwise, the residual version of networks was used.

The L4 and L8 networks were trained on mini-batches of 64 64 × 64 patches and 4 128 × 128 patches respectively. The patches were randomly sampled from training images. The number of training iterations was fixed to 250 K which is significantly less compared to AR-CNN's 10^7 iterations. The learning rate was scaled down by factor of 2 every 50 K iterations. The networks were initialized by the Xavier initialization [10] in the first three layers, and a Gaussian initialization with lower variation was used in the final layer. The learning rate of the last layer was set ten times smaller than for the other layers.

Artifacts reduction quality. The results of artifacts removal on LIVE1 dataset with JPEG quality 10 and 20 are shown in Table 3. The results on the BSDS500 validation set are presented in Table 4. L8 outperforms all other methods with significantly higher scores in all three quality measures. L4 which performs worse compared to L8, still surpasses the other methods in most cases even though it is much small and computationally efficient compared to L8. Examples of resulting images are presented in Figure 4.

JPEG quality generalization. We evaluated the ability of the trained networks to generalize to a different compression quality by training L4 on one quality and evaluating on other qualities (L4Q10 trained for quality 10, L4Q20 for quality 20, etc up to L4Q60). To asses the ability of CNNs to handle multiple compression qualities in a single model, we trained a single L4 network on all the qualities together (L4Q10-Q60). The results in Figure 5 show that L4Q10-Q60 provides stable results across the quality range. However, the quality-specific networks perform better for their respective qualities. The quality-specific networks generalize only to similar qualities. In practice, a single network should easily be able to handle smaller quality ranges (e.g. 10–20 quality points wide) when trained on data from the whole range.

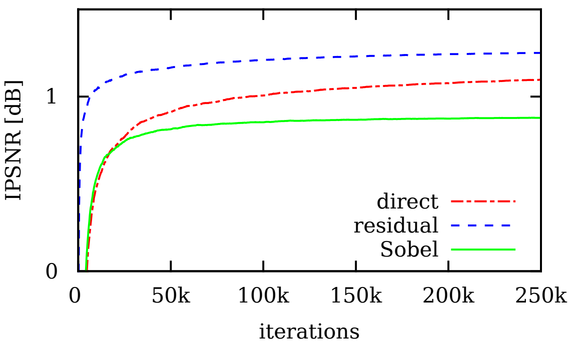


Figure 6: Training development of L4 with different training objectives.

Impact of learning objective. We compare L4 networks trained for direct mapping, residual, and edge preserving loss. Although the architecture and initialization of all the L4 networks were the same, we had to select suitable learning rates (lr) and weight decay coefficients (wd) by performing grid search for each learning objective separately. The chosen values are for direct mapping lr 0.4, wd 5×10^{-7}, for residual learning lr 8, wd 5×10^{-7}, and for edge preserving objective lr 0.05, wd 5×10^{-4}.[2] The values were chosen on JPEG quality 10 and they were used for all other qualities.

The progress of learning is shown in Figure 6. The residual network converges much faster compared to the direct mapping network. The results on LIVE1 measured by PSNR, PSNR-B and SSIM are in Table 5.

Figure 7 shows 1st layer filters of the networks during different stages of training. All the networks formed reasonable-looking filters. The residual network

Objective	PSNR	PSNR-B	SSIM
Distorted	27.58	24.97	0.769
Direct mapping	28.99	28.66	0.820
Edge preserving	28.69	28.40	0.813
Residual learn.	**29.08**	**28.71**	**0.824**

Table 5: Results of L4 networks with different objectives on LIVE1 dataset with quality 10.

[2] In our experiments, the loss was normalized by the number of output pixels. This scaling influences the scale of gradients and results in relatively high learning rates and low weight decay coefficients.

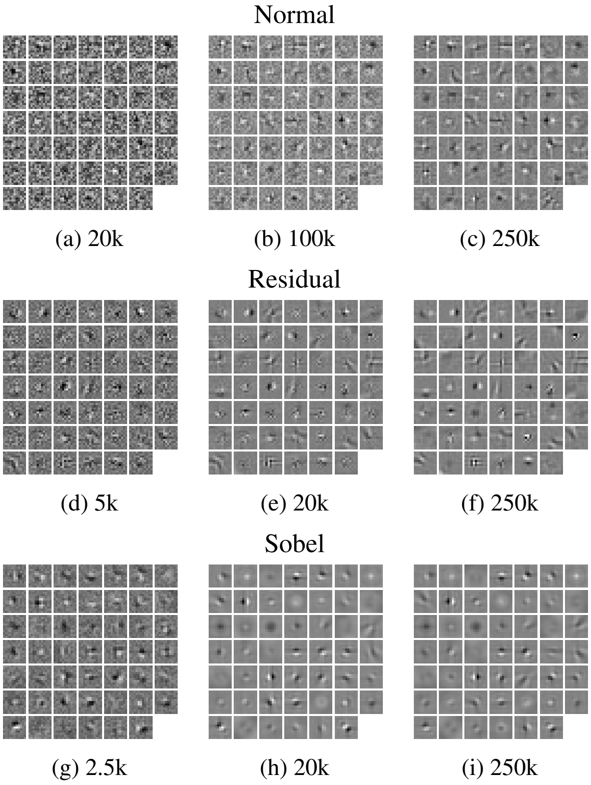


Figure 7: Filters from the first layer of L4 networks with normal/residual/Sobel (edge preserving) objective at different stages of training. Iterations are showed below the images.

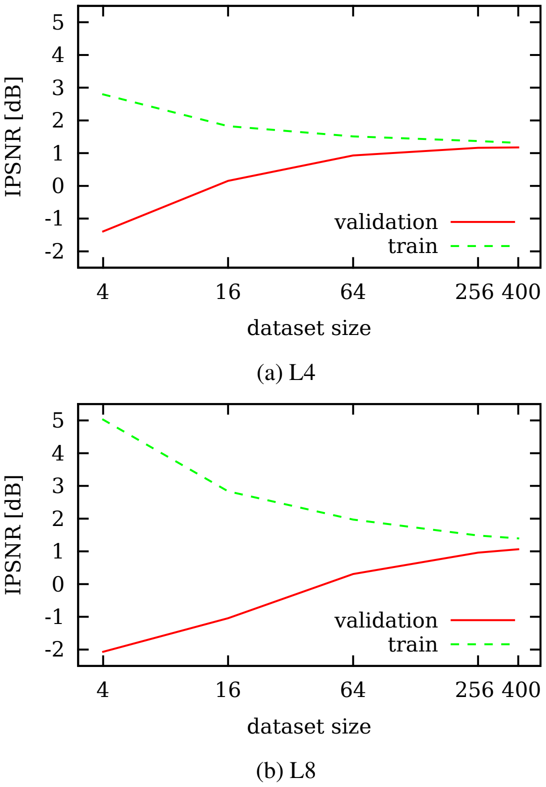


Figure 8: Generalization for different sized train set.

formed more complex higher frequency filters compared to the other networks. The edge preserving network learned a number of low-pass filters which are probably needed to transfer the general image appearance through the network – these filters are missing in the residual network. The filters of the normal direct mapping network remain noisy, which could be due to different weight decay coefficient the low learning rate, or their combination.

The results show that the residual learning is beneficial for JPEG artifact reduction in terms of resulting reconstruction quality and training speed. On the other hand, the edge preserving objective does not improve resulting quality noticeably in the case of L4.

Dataset size. The quality of reconstruction achieved by larger networks may suffer due to inadequate size of a training set. In order to asses how the L4 and L8 behave with respect to training set size, we trained the residual versions of the networks on 4, 16, 64, 256, and 400 images from the training set. The L4 and L8 networks contain approx 70 K and 220 K learnable parameters respectively which suggests that L8 should require larger training set for the same generalization. Figure 8 shows results on the different training sets and corresponding results on the independent test set. Both networks clearly overfit on the smaller datasets. L8 overfits significantly more and it would require more images to reach proper generalization, while L4 seems to reach perfect generalization already on the relatively small dataset of 400 images.

Speed. Using cuDNN v3 implementation of convolutions on GeForce GTX 780, we were able to process 1 Mpx images in 220 ms with network L4 and in 1052 ms with L8. The L4 and L8 networks require approximately 140 K and 440 K floating point operations per pixel, respecively.[3]

5 CONCLUSIONS

In this work, we show that it is possible to train large and deep networks for JPEG artifacts removal which outperform previous state-of-the-art results of smaller networks. We combine the residual learning by Kim *et al.* [16], skip architecture [18], and symmetric weight initialization which allowed us to successfully train networks with 8 layers.

We compare networks with three different objectives – direct mapping, residual learning, and edge preserving. The best reconstruction results are provided by the residual learning.

[3] The networks, processed images, and implementations will be available at http://www.fit.vutbr.cz/ ihradis/CNN-Deblur/.

We further investigate the network ability to generalize across different compression JPEG quality levels. Our results show that it is possible to use one network trained for several qualities as an acceptable trade-off.

Finally, we evaluate generalization of the networks with respect to training set size. The results suggest that small networks similar to L4 (20 K parameters) can be safely trained on the BSD dataset. However, the generalization of L8 (100 K parameters) and larger networks is not guaranteed on this small dataset and a larger common dataset should be compiled to allow fair and consistent evaluation in the future.

In a future work, we intend to apply convolutional networks to other compression methods, for example, JPEG 2000, JPEG XR, or WebP. Next, we would like to train convolutional networks to reconstruct images directly from the JPEG coefficients which should provide the networks with significant clues as to which image elements are and which are not artifacts. The receptive field even of the L8 network is still relatively small and we expect that it should be possible to reach higher reconstruction quality by increasing the receptive field or by providing context information by other means.

Acknowledgements

This work has been supported by the ARTEMIS joint undertaking under grant agreement ALMARVI (no. 621439), the Technology Agency of the Czech Republic (TA CR) Competence Centres project V3C – Visual Computing Competence Center (no. TE01020415) and the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science (no. LQ1602).

REFERENCES

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5): 898–916, May 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.161.

[2] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2 (1):1–127, 2009. doi: 10.1561/2200000006. Also published as a book. Now Publishers, 2009.

[3] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399, June 2012. doi: 10.1109/CVPR.2012.6247952.

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science (LNCS) 8692, pages 184–199. Springer International Publishing, Sept. 2014. ISBN 978-3-319-10593-2. doi: 10.1007/978-3-319-10593-2_13. Part IV.

[5] C. Dong, Y. Deng, C. C. Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *International Conference on Computer Vision (ICCV)*, pages 576–584, 2015.

[6] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *International Conference on Computer Vision (ICCV)*, pages 633–640, 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.84.

[7] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise Shape-Adaptive DCT for high-quality deblocking of compressed color images. In *European Signal Processing Conference (EUSIPCO)*, Sept. 2006.

[8] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise Shape-Adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Transactions on Image Processing*, 16 (5):1395–1411, May 2007.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. to appear in CVPR 2016.

[12] M. Hradis, J. Kotera, P. Zemcik, and F. Sroubek. Convolutional neural networks for direct text deblurring. In X. Xie, M. W. Jones, and G. K. L. Tam, editors, *British Machine Vision Conference (BMVC)*, pages 6.1–6.13. BMVA Press, Sept. 2015. ISBN 1-901725-53-7. doi: 10.5244/C.29.6.

[13] International Telegraph and Telephone Consultative Committee. CCITT recommendation T.81: Terminal equipment and protocols for telematic services : Information technology - digital compression and coding of continuous-tone still images - requirements and guidelines, 1993.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In D. Blei and F. Bach, editors, *International Conference on Machine Learning (ICML)*, pages 448–456. JMLR Workshop and Conference Proceedings, 2015.

[15] V. Jain and S. Seung. Natural image denoising with convolutional networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, pages 769–776. Curran Associates, 2009.

[16] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. *CoRR*, abs/1511.04587, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, 2012.

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *Computer Vision and Pattern Recognition (CVPR)*, Nov. 2015.

[19] A. Nosratinia. Embedded post-processing for enhancement of compressed images. In *Data Compression Conference (DCC)*, pages 62–71, 3 1999. doi: 10.1109/DCC.1999.755655.

[20] C. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2481418.

[21] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[22] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. LIVE image quality assessment database release, 2015.

[23] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Computer Vision and Pattern Recognition (CVPR)*, pages 769–777, June 2015. doi: 10.1109/CVPR.2015.7298677.

[24] P. Svoboda, M. Hradis, L. Marsik, and P. Zemcik. CNN for license plate motion deblurring. *CoRR*, abs/1602.07873, 2016.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, June 2014. doi: 10.1109/CVPR.2014.220.

[26] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, June 2014. doi: 10.1109/CVPR.2014.214.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 4 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861.

[29] T. S. Wong, C. A. Bouman, I. Pollak, and Z. Fan. A document image model and estimation algorithm for optimized JPEG decompression. *IEEE Transactions on Image Processing*, 18(11):2518–2535, Nov. 2009. ISSN 1057-7149. doi: 10.1109/TIP.2009.2028252.

[30] L. Xu, J. S. J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014.

[31] S. Yang, S. Kittitornkun, Y.-H. Hu, T. Q. Nguyen, and D. L. Tull. Blocking artifact free inverse discrete cosine transform. In *International Conference on Image Processing*, volume 3, pages 869–872, 2000. doi: 10.1109/ICIP.2000.899594.

[32] C. Yim and A. C. Bovik. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, Jan. 2011. ISSN 1057-7149. doi: 10.1109/TIP.2010.2061859.