Preserving Low-Quality Video through Deep Learning

Leonardo Galteri, Lorenzo Seidenari, Tiberio Uricchio, Marco Bertini, Alberto Del Bimbo

University of Florence, Italy

E-mail: {name.lastname}@unifi.it

Abstract. Lossy video stream compression is performed to reduce the bandwidth and storage requirements. Moreover also image compression is a need that arises in many circumstances.It is often the case that older archive are stored at low resolution and with a compression rate suitable for the technology available at the time the video was created. Unfortunately, lossy compression algorithms cause artifact. Such artifacts, usually damage higher frequency details also adding noise or novel image patterns. There are several issues with this phenomenon. Low-quality images can be less pleasant to persons. Object detectors algorithms may have their performance reduced. As a result, given a perturbed version of it, we aim at removing such artifacts to recover the original image. To obtain that, one should reverse the compression process through a complicated non-linear image transformation. We propose a deep neural network able to improve image quality. We show that this model can be optimized either traditionally, directly optimizing an image similarity loss (SSIM), or using a generative adversarial approach (GAN). Our restored images have more photorealistic details with respect to traditional image enhancement networks. Our training procedure based on sub-patches is novel. Moreover, we propose novel testing protocol to evaluate restored images quantitatively. Differently from previously proposed approaches we are able to remove artifacts generated at any quality by inferring the image quality directly from data. Human evaluation and quantitative experiments in object detection show that our GAN generates images with finer consistent details and these details make a difference both for machines and humans.

1. Introduction

A huge number of videos are produced, streamed and shared on the web, and many more are used within private systems, such as mobile phones, cameras and surveillance systems. To store efficiently and transmit these videos compression is necessary. This allows to reduce bandwidth and storage. Compressions is tipically lossy, given the need to deal with large quantities of data, such as HD and 4K resolutions which are more and more common. These algorithms application results in a more or less strong loss of content quality, to achieve a better compression ratio. However, compression algorithms are designed to reduce loss of perceptual quality, exploiting some human visual system mathematical model.

Compression of videos causes artifacts to appear. Artifacts are due to the different types of lossy compressions used. MPEG-based algorithms such as H.264 and H.265/AVC or AV1, are the most common and recent algorithm used nowadays. In such case artifacts are due to sub-sampling of chroma (i.e. dropping of color information) and the DCT coefficient quantization; Due to how the original frame is partitioned blocking artifacts also arise. Blocking artifacts are also due to erroneous motion compensated prediction [18]. Lossy image compression algorithms such as JPEG share similar artifacts. Finally, some artifacts are caused by erroneous motion compensation and coding, yielding flickering; this is caused by differences in frame reconstruction between intra-frames and inter-frames (i.e. key frames encoded as images and frames reconstructed using motion compensation) [30].

2. Related Work

Image enhancement has been vastly studied in the past, especially in the case of compressed media. Several techniques are based on image processing algorithms [3, 6, 13, 16, 17, 29, 31, 33, 34]. Very recently, learning methods have been developed [15, 4, 20, 27, 28, 8, 7]. Deep Convolutional Neural Networks (DCNN), trained to restore image quality using couples of undistorted and distorted images, obtain the best quality. A major strength such approaches is that knowing the process of image degradation, training data can be generated automatically without the need for hand labeling. Degraded images are used as input to restoration networks while high quality sources are used as target images. Kang *et al* [15] address both super-resolution and deblocking in the case of highly-compressed images, learning sparse representations that model the relationship between low- and high-resolution image patches with and without blocking artifacts. The approach is tested on highly compressed JPEG images, with QF values between 15 and 25.

Artifact removal using deep learning was first addressed by Dong *et al* [4] extending their work on super-resolution SRCNN. A similar method has been proposed by Svoboda *et al* [27] with improved results. Such improvement with respect to [4] is due to the more complex architecture using skip-connections and residual learning. All weights are learned and there are no specific functions implemented by the architecture. Residual blocks are used in many recent works [2, 7, 32], favoring deeper architectures. In Cavigelli *et al* [2], a deep residual architecture with 12 layers and hierarchical skip-connections is used. Yoo*et al* used a local frequency classifier to condition and encoder-decoder network for JPEG artifact removal. In our previous work we propose a GAN ensemble driven by a quality classifier allowing to restore static images of unknown quality[7]. Perceived is superior to competing approaches, thanks to adversarial training.

To the best of our knowledge the only method restoring compressed video frames is proposed by He *et al* [12]. Although their method is tightly bound to HEVC coding. It exploits information from coding units to learn a two-stream convolutional network which receives a decoded frame and combines it with a feature map computed from the partition data.

3. Methodology

The objective of video compression artifact removal is to obtain a reconstructed frame image I^R from a compressed input image I^C. In this context, $I^C = C(I)$ is the output frame of a video compression algorithm C and I is an uncompressed input frame. Different C algorithms will result in different I^C frames, with different compression artifacts. Many image and video compression algorithms (e.g. JPEG, JPEG2000, VP9, H.264/AVC, H.265/HEVC) use color spaces that separate luminance from chrominance information, like YCrCb. This allows to better de-correlate color components leading to a more efficient compression; it also permits a first step of lossy compression through chrominance sub-sampling, based on the fact that the human visual system has reduced sensitivity to its variations.

We represent frames I^R, I^C and I as real valued tensors with dimensions $W \times H \times Ch$, where W and H are width and height, respectively, and Ch is the number of color channels. In cases where the quality assessment is performed using luminance only we transform images to gray-scale considering only the Y channel, and $Ch = 1$, in all other cases we have $Ch = 3$, using the RGB color space as is commonly done when working with CNNs.

The compression of an uncompressed video frame $I \in [0, 255]^{W \times H \times Ch}$ is performed according to:

$$I^C = C\left(I, QF\right) \in [0, 255]^{W \times H \times Ch} \tag{1}$$

using a function C, representing some video compression algorithm, which is parametrized by some quality factor QF. The problem of video compression artifacts removal can be seen as to compute an inverse generator function $G \approx C_{QF}^{-1}$ that reconstructs I from I^C:

$$G\left(I^C\right) = I^R \approx I \tag{2}$$

Each generator can in principle be trained with images obtained from different QFs. In practice we have shown in [9], that single QF generators perform better and can be driven by a QF predictor.

To learn the generator function, we train a convolutional neural network $G\left(I^C; \theta_g\right)$ where $\theta_g = \{W_{1:K}; b_{1:K}\}$ are the parameters representing weights and biases of the K layers of the network. Given N training images we optimize a custom loss function l_{AR} by solving:

$$\hat{\theta}_g = \arg\min_{\theta_g} \frac{1}{N} \sum_{n=1}^{N} l_{AR}\left(I, G\left(I^C, \theta_g\right)\right) \tag{3}$$

In principle, the elimination of compression artifacts is a task that can be classified as image transformation problem; it comprises several other tasks from super-resolution to style-transfer. Recent works have shown that this category of tasks can be conveniently solved using generative approaches, i.e. learning a fully convolutional neural network (FCN) [19] that given a certain input image is able to generate, as an output, an improved version of it. A motivation to use FCN architectures for these image processing tasks is that they are extremely convenient to perform local non-linear image transformations; moreover, thanks to the lack of fully connected layers they can process images and video frames of any size. This property is advantageous to speed up the training process. In fact, the artifacts that we want to remove typically appear at scales close to the block size used by the compression algorithm C. For this reason we can learn models on smaller patches using larger batches.

Therefore, we propose to use fully convolutional architecture that can be optimized both with direct supervision or otherwise combined in a generative adversarial framework using a novel discriminator. Details of the proposed networks are presented in the following, together with the proposed loss functions.

3.1. Generative Network

We use a deep residual network as a generator. The architecture is mainly composed by convolutional layer blocks with LeakyReLU activations.

Our generator is inspired by [11]. Layers have 64 3×3 convolution kernels. After the first convolutional layer we use stride to half the size of feature maps. We then use 15 residual blocks with a padding of 1 pixel. The final upsampling is performed with a nearest-neighbour approach followed by a final convolutional layer to remove upsampling artifacts [24]. The output image is generated by a convolutional layer with a *tanh* activation. This produces output tensors with values in $[-1, 1]$, which are therefore comparable to the rescaled image input.

3.1.1. Discriminative Network The discriminator is a sequence of convolutional layers with one pixel stride and no padding. LeakyReLU activations are used after each layer. Filter amount is doubled every two layers. We do not use fully connected layers to allow different input resolution patches. Each pixel of the final output is computed with a sigmoid activation, being

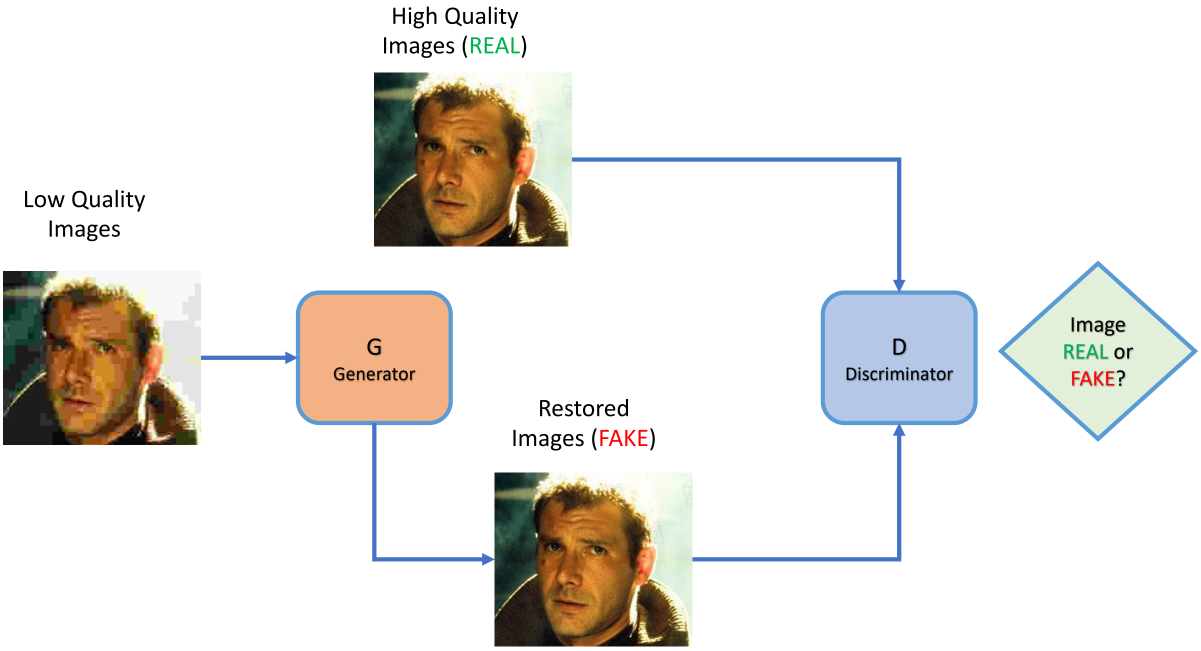


Figure 1. Generative Adversarial Network trained for image artifact removal. Low quality input images are the input of the generator. The Generator tries to "fool" the discriminator network with restored images. The discriminator is trained to tell apart real from fake images.

the discriminator solving a binary classification problem. Average pooling is used to aggregate decisions over pixels.

The set of weights ψ of the D network are learned by minimizing:

$$l_d = -\log\left(D_\psi\left(I|I^C\right)\right) - \log\left(1 - D_\psi\left(I^R|I^C\right)\right) \tag{4}$$

We propose to discriminate at the sub-patch level; thanks to this approach we can better detect patterns arising from removing artifacts which are formed into fine structures, typically 16×16 patches.

3.1.2. Perceptual Loss We design our loss exploiting findings from Dosovitskiy and Brox [5], Johnson *et al.* [14], Bruna *et al* [1] and Gatys *et al* [10]; we add a perceptual similarity loss to the adversarial loss. In particular, The distance between images is computed after projecting I and I^R on a feature space by some differentiable function ϕ and taking the Euclidean distance between the two feature representations:

$$l_P = \frac{1}{W_f H_f} \sum_{x=1}^{W_f} \sum_{y=1}^{H_f} \left(\phi\left(I\right)_{x,y} - \phi\left(I^R\right)_{x,y}\right)^2 \tag{5}$$

where W_f and H_f are respectively the width and the height of the feature maps. The effect obtained using this loss is that output images will be closer to input high quality images not in the pixel space but in the feature space of some neural network.

In this work we compute $\phi\left(I\right)$ by extracting the feature maps from a pre-trained VGG-19 model [26], using the second convolution layer before the last max-pooling layer of the network, namely `conv5_3`.

3.1.3. Adversarial Patch Loss We train the generator combining the perceptual loss with the adversarial loss thus obtaining:

$$l_{AR} = l_P + \lambda l_{adv}. \tag{6}$$

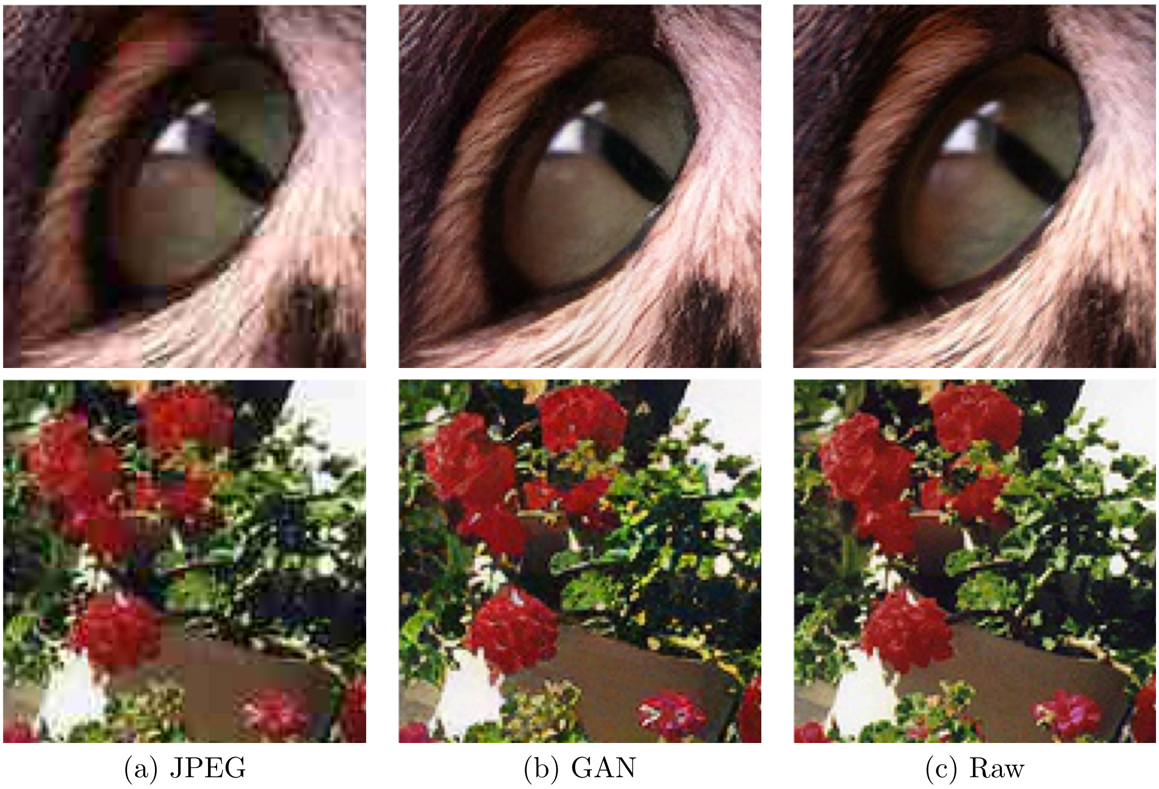
Where l_{adv} is the standard adversarial loss:

$$l_{adv} = -\log\left(D_\psi\left(I^R | I^C\right)\right) \tag{7}$$

that rewards solutions that are able to "fool" the discriminator.

4. Experiments

First we show qualitative results for static content. Our algorithm can be trained to restore quality of several lossy compressed images; here for the sake of simplicity we only show restoration on JPEG compressed images. In Fig. 2 we show the restoration of highly compressed content on two patches. A first mean to measure quality of images is to apply so called no-reference metrics. We use two vastly used algorithms [21, 23]. Both algorithms are based on the concept of image naturalness and while [21] trains a regressor to predict image quality scores provided by humans based on image statistics, [23] is based on the idea of measuring the similarity between the statistics of features of highly scored images and the images under inspection. Resutls are reported in Tab.1.



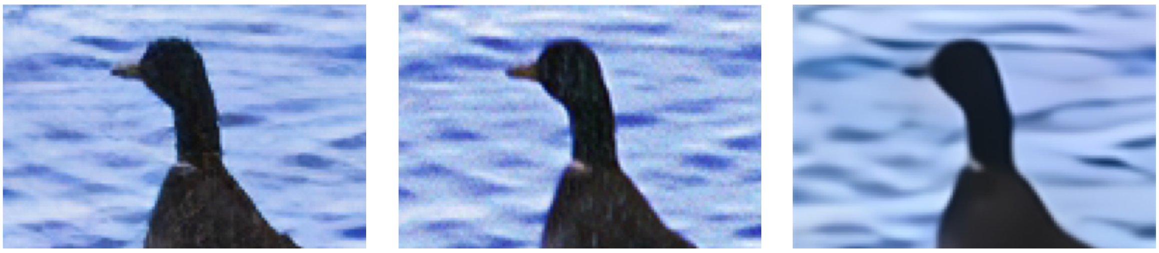
(a) JPEG (b) GAN (c) Raw

Figure 2. Our reconstruction algoritm (b) applied on JPEG compressed images (a) compared to RAW images (c). Our method is able to remove all blocking artifacts and also keeping high frequency details such as cat's fur or the shape of rose's petals.

We now report results on video frames. Here frames are degraded by H.264 compression. In Fig. 3. We show how our method compares with Wave.one algorithm[25]. While we have slightly larger videos (3×), measured in bits per pixel(bpp) our algorithm has a much higher quality and runs much faster(12×). In Table 2 we report an analysis based on no-reference metrics applied to video frames. To take into account the peculiar nature of video we also report results of the VIIDEO[22] algorithm, which evaluate also the temporal nature of media.

	NIQE[23]	BRISQUE[21]
JPEG10	6.36	53.17
GAN	**4.27**	**19.65**
ORIG	4.35	24.32

Table 1. Evaluation of our method using popular no-reference metrics, lower is better. GAN score computed by both metrics is better than the score of ORIG.



Our: bpp 0.146
FPS: 20

Raw: bpp 12
FPS: -

WaveOne [25]: bpp 0.0570
FPS: 1.6

Figure 3. Comparison of our method with [25]. Our network is faster (12x) and although uses more bpp (3x) has much better quality while [25] is overly smoothed.

	VIIDEO[22]	NIQE [23]	BRISQUE[21]	FPS@720p
H.264	0.520	4.890	41.93	-
Our Very Fast	0.388	4.574	25.12	42
Our Fast	**0.350**	3.714	**16.95**	20
Galteri *et al* [7]	0.387	**3.594**	17.58	4
Uncompressed	0.276	4.329	23.73	-

Table 2. No reference quality assessment of our compression artifact removal networks. VIIDEO is specifically designed for sequences, while NIQE and BRISQUE are geared towards images. For all metrics lower figure is better.

In Fig. 4 are reported subjective evaluation results as MOS (Mean Opinion Scores) as box plots, showing the quartiles of the scores (box), while the whiskers show the rest of the distribution. The plots are made for the original images, the images compressed with JPEG using a QF=10, and the images restored with our GAN-based approach from the heavily compressed JPEG images. The figure shows that the GAN-based network is able to produce images that are perceptually of much higher quality than the images from which they are originated; the average MOS score for JPEG images is 1.15, for our GAN-based approach is 2.56 and for the original images it is 3.59. The relatively low MOS scores obtained also by the original images are related to the fact that COCO images have a visual quality that is much lower than that of dataset designed for image quality evaluation.

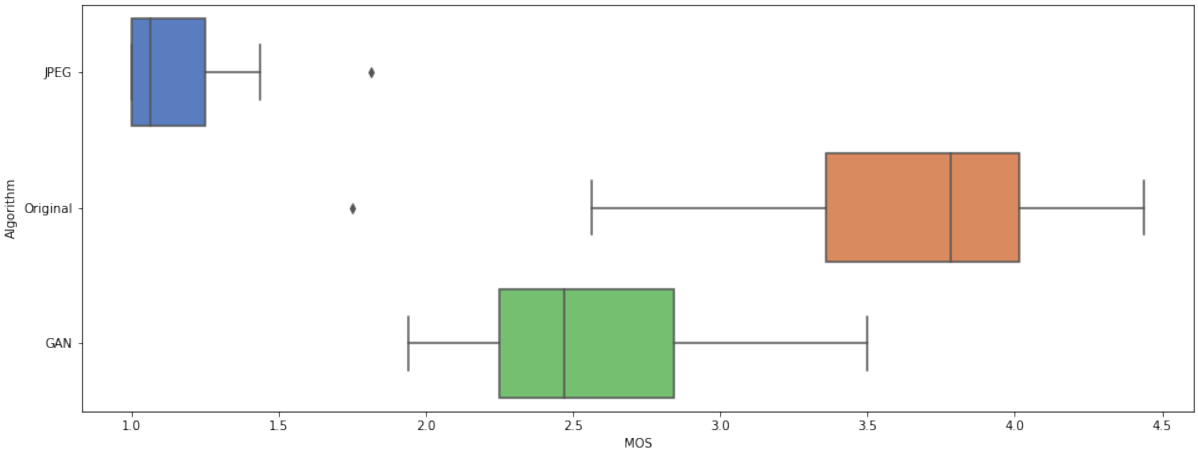


Figure 4. Subjective image quality evaluation of original COCO images (orange), heavily compressed JPEG images (blue) and their restored version obtained with our GAN-based approach (green). Restored images are perceived as having abetter quality than their compressed versions.

5. Conclusion

In this paper we have shown a methodology to recover quality from video and image archive that may have been preserved digitally at a low quality. Image and video storage is often pursued via lossy compression which causes artifacts. We propose to use deep generative adversarial network to recover quality of existing video archives. Our results show that our algorithm improves quality with respect to exisiting no reference metrics and according to human evaluators. Interestingly our algorithm can also be executed in real-time on modern GPUs, we expect that in the future the improvement of hardware support for neural network would allow our method to be applied directlu on consumer hardware such as smartphones and smartTVs.

References

[1] Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. *CoRR*, abs/1511.05666, 2015.

[2] Lukas Cavigelli, Pascal Hager, and Luca Benini. CAS-CNN: A deep convolutional neural network for image compression artifact suppression. In *Proc. of IJCNN*, 2017.

[3] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes. Postprocessing of compressed images via sequential denoising. *IEEE Transactions on Image Processing*, 25(7):3044–3058, July 2016.

[4] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proc. of ICCV*, 2015.

[5] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. of NIPS*, 2016.

[6] Alessandro Foi, Vladimir Katkovnik, and Karen 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, 2007.

[7] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo. Deep universal generative adversarial compression artifact removal. *IEEE Transactions on Multimedia*, pages 1–1, 2019.

[8] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Deep generative adversarial compression artifact removal. In *Proc. of ICCV*, 2017.

[9] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Deep universal generative adversarial compression artifact removal. *IEEE Transactions on Multimedia (TMM)*, 21(8):2131–2145, Aug 2019.

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *CoRR*, abs/1505.07376, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of CVPR*, 2016.

[12] Xiaoyi He, Qiang Hu, Xiaoyun Zhang, Chongyang Zhang, Weiyao Lin, and Xintong Han. Enhancing HEVC compressed videos with a partition-masked convolutional neural network. In *Proc. of ICIP*, 2018.

[13] V. Jakhetiya, W. Lin, S. P. Jaiswal, S. C. Guntuku, and O. C. Au. Maximum a posterior and perceptually motivated reconstruction algorithm: A generic framework. *IEEE Transactions on Multimedia*, 19(1):93–106, 2017.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. of ECCV*, 2016.

[15] L. W. Kang, C. C. Hsu, B. Zhuang, C. W. Lin, and C. H. Yeh. Learning-based joint super-resolution and deblocking for a highly compressed image. *IEEE Transactions on Multimedia*, 17(7):921–934, 2015.

[16] Tao Li, Xiaohai He, Linbo Qing, Qizhi Teng, and Honggang Chen. An iterative framework of cascaded deblocking and super-resolution for compressed images. *IEEE Transactions on Multimedia*, 2017.

[17] Yu Li, Fangfang Guo, Robby T. Tan, and Michael S. Brown. A contrast enhancement framework with JPEG artifacts suppression. In *Proc. of ECCV*, 2014.

[18] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz. Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):614–619, 2003.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of CVPR*, 2015.

[20] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Proc. of NIPS*, 2016.

[21] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, Dec 2012.

[22] Anish Mittal, Michele A Saad, and Alan C Bovik. A completely blind video integrity oracle. *IEEE Transactions on Image Processing*, 25(1):289–300, 2016.

[23] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a completely blind image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

[24] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. http://distill.pub/2016/deconv-checkerboard.

[25] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G. Anderson, and Lubomir Bourdev. Learned video compression, 2018.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR*, 2015.

[27] Pavel Svoboda, Michal Hradis, David Barina, and Pavel Zemcik. Compression artifacts removal using convolutional neural networks. *arXiv preprint arXiv:1605.00366*, 2016.

[28] Zhangyang Wang, Ding Liu, Shiyu Chang, Qing Ling, Yingzhen Yang, and Thomas S Huang. D3: Deep dual-domain based fast restoration of JPEG-compressed images. In *Proc. of CVPR*, 2016.

[29] Tak-Shing Wong, Charles A Bouman, Ilya Pollak, and Zhigang Fan. A document image model and estimation algorithm for optimized JPEG decompression. *IEEE Transactions on Image Processing*, 18(11):2518–2535, 2009.

[30] J. X. Yang and H. R. Wu. Robust filtering technique for reduction of temporal fluctuation in h.264 video sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(3):458–462, March 2010.

[31] Seungjoon Yang, Surin Kittitornkun, Yu-Hen Hu, Truong Q Nguyen, and Damon L Tull. Blocking artifact free inverse discrete cosine transform. In *Proc. of ICIP*, 2000.

[32] Jaeyoung Yoo, Sang-ho Lee, and Nojun Kwak. Image restoration by estimating frequency distribution of local patches. In *Proc. of CVPR*, 2018.

[33] J. Zhang, R. Xiong, C. Zhao, Y. Zhang, S. Ma, and W. Gao. CONCOLOR: Constrained non-convex low-rank model for image deblocking. *IEEE Transactions on Image Processing*, 25(3):1246–1259, March 2016.

[34] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao. Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. *IEEE Transactions on Image Processing*, 22(12):4613–4626, 2013.