Towards Real-Time Image Enhancement GANs

Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo

Università di Firenze - MICC, Firenze, Italy
[name.surname]@unifi.it
http://www.micc.unifi.it/

Abstract. Video stream compression, using lossy algorithms, is performed to reduce the bandwidth required to transmit them. Typical cases in which this may happen are commercial streaming services, video surveillance or camera enabled IoT devices. In all these cases a solution to improve the video quality, either for human view or for automatic video analysis, is to perform a post-processing that eliminates the compression artifacts introduced by the compression algorithms.
Generative Adversarial Network have been shown to obtain extremely high quality results in image enhancement tasks; however, to obtain top quality results high capacity large generators are usually employed, resulting in high computational costs and processing time. In this paper we present an architecture that can be used to reduce the cost of generators, paving a way towards real-time frame enhancement with GANs.
With the proposed approach, enhanced images appear natural and pleasant to the eye. Locally high frequency patterns often differ from the raw uncompressed images. A possible application is to improve video conferencing, or live streaming. In these cases there is no original uncompressed video stream available. Therefore, we report results using popular no-reference metrics showing high naturalness and quality even for efficient networks.

1 Introduction

Every day a huge number of videos are produced, streamed and shared on the web, and many more are produced and used within private systems and networks, such as mobile phones, cameras and surveillance systems. To practically store and transmit these video streams it is necessary to compress them, to reduce bandwidth and storage requirements. Typically video is compressed using lossy algorithms, given the need to deal with large quantities of data, especially when dealing with HD and 4K resolutions that are becoming more and more common. The effect of these algorithms results in a more or less strong loss of content fidelity with respect to the original visual data, to achieve a better compression ratio. However, since one of the factors that accounts for user experience is image quality, compression algorithms are designed to reduce perceptual quality loss, according to some model of the human visual system.

A typical use case in which a high compression is desirable is that of video conferencing, in which video streams must be kept small to reduce communication latency and thus improve user experience. Another case is that of wireless cameras, in particular mobile and wearable ones, that may need to limit power consumption reducing the energy cost of image transmission applying strong compression. Also in the case of entertainment video streaming, like Netflix or Amazon Prime Video, there is need to reduce as much as possible the required bandwidth, to reduce network congestion and operational costs. Video conferencing and video surveillance require to spare bandwidth, adding also the requirement of reduced latency, that can be achieved through compression. Another use case is that of videos encoded and transmitted from IoT devices and drones, in which there is the low-energy constraint that calls for high compression of the streams to reduce transmission time, to save battery power.

When compressing videos several artifacts appear as shown in Fig. 1. These artifacts are due to the different types of lossy compressions used. Considering MPEG-based algorithms such as H.264 and H.265/AVC or AV1, the most common and recent algorithm used nowadays, these artifacts are due to the chroma sub-sampling (i.e. dropping some color information of the original image) and the quantization of the DCT (or DST) coefficients; other artifacts are due to blocking processing of frames, due to how the original uncompressed frame is partitioned for compression, but is also due to erroneous motion compensated prediction [20]. These two causes are common also in image compression algorithms such as JPEG. Finally, another source of artifacts is associated with motion compensation and coding, such as flickering, 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) [31].

An approach to improve the perceived image quality while maintaining a high compression rate is to perform filtering on the reconstructed frames, to reduce the effect of the various artifacts. The most recent codecs, such as H.265 and AV1 envisage standardised deblocking filtering.

In this work we propose a solution to artifact removal based on convolutional neural networks trained on large sets of frame patches compressed with different quality factors. Our approach is independent with respect to the compression algorithm used to process a video; it can be used as a post-processing step on decompressed frames and therefore it can be applied on many lossy compression algorithms such as WebM, AV1, H.264/AVC and H.265/HEVC.

One of the main advantages of improving video quality working on artifact removal is that our method can be applied just on the receiving end of a coding pipeline, thus avoiding any modification to the existing compression pipelines, that are often optimized e.g. using dedicated hardware such as graphic cards, GPUs or SoCs. Another important aspect is that often streaming services use a dynamic adaptive streaming approach, that compose video streams using different versions encoded at different bit rates (and thus at different qualities) to cope with bandwidth availability; an example of this is the Dynamic Adaptive Streaming over HTTP (DASH) [27]. This means that we could not rely on an approach based on super-resolution, that would require image sub-sampling also on the coding end.

2 Related Work

Improving image quality is a topic that has been vastly studied in the past, especially in the case of compression artifact removal. Many approaches are based on image process-

Fig. 1: Examples of video compression artifacts: *top*) blocking artifacts on the bottom part of the green pole, color interpolation due to chroma sub-sampling on left and right of the umbrella; *mid*) and *bottom*) ringing around the white on blue text in the street name sign; little vertical flickering (1-2 lines) of numbers and letters (in particular N and 7) between two following frames.

ing techniques [5, 8, 14, 18, 19, 30, 32, 34, 35]. Recently, several learning based methods have been proposed [6, 9, 10, 16, 21, 28, 29].

Best results have been obtained using Deep Convolutional Neural Networks (DCNN), trained to restore image quality using couples of undistorted and distorted images. A major strength of this type of approach is that once the degrading process is known, e.g. video compression, generating data is extremely easy and does not requires hand labeling, but just to generate degraded images from high quality input. Degraded images will be fed as input to restoration networks while high quality sources will be regarded as ground truth or target images. Dong *et al.* [6] extended their previous work on super-resolution SRCNN with artifact removal CNN (AR-CNN) sharing a common architecture with SRCNN, following sparse coding pipelines. Svoboda *et al.* [28] follow a similar approach, obtaining improved results in image restoration by learning a feed-forward CNN. The main difference with respect to [6] is that the CNN is a standard multi-layered network in which the layers have no specific functions, using residual learning and skip connections. Recent works [3,9,33] tend to use deeper architectures, often employing residual blocks. Cavigelli *et al.* [3] trained 12-layers CNN to remove JPEG artifacts. Their CNN uses skip connection in a hierarchical fashion. Local frequency classifier are trained and exploited by Yoo*et al.* to condition and encoder-decoder architecture in reconstructing JPEG compressed images [33]. Galteri *et al.* are the first to propose a method that uses a GAN ensemble and a quality predictor that allows them to restore images of unknown quality [9]. Their method shows superior results in term of perceived quality.

To the best of our knowledge the only method restoring compressed video frames is proposed by He *et al.* [13]. Their method, tightly bound to HEVC coding, exploits coding unit to learn a two-stream CNN receiving a decoded frame and combines it with a feature map computed from the partition data.

2.1 Contribution

Unfortunately all best performing artifact removal methods do not perform in real-time, even on high-end GPUs. All the envisioned applications for such technology in video quality improvement have a less or more strict real-time constraint. In this paper we propose a GAN in which the generator is designed with efficiency in mind. Instead, the discriminator can be made large at will, since it only affects training efficiency. We show that on no-reference video and image quality assessment our approach produces frames that have scores higher than compressed frames.

3 Methodology

Our approach consists in regarding a frame from a compressed video as an image which has been distorted by some known process. Considering a raw frame I_t from a sequence we consider $I_t^C = C(I_t, I_{t-1}, I_{t-2}, \ldots, \theta)$ as its compressed version, where $C(\cdot)$ is some compression algorithm for video sequences such as H.264/AVC configured according to a parameters set θ.

Representing images as real valued tensors in $\mathbb{R}^{H \times W \times Ch}$, where W and H are width and height of the frame, and Ch is the number of channels, we would like to learn some function $G(\cdot)$ able to invert the compression process:

$$G\left(I_t^C\right) = I_t^R \approx I_t \tag{1}$$

where I_t^R denotes the restored version of I_t^C. In all models we use $Ch = 3$, since we are training over and restoring RGB video frames. We define the function $G(\cdot)$ as a fully convolutional neural network, whose weights are learned using a Generative Adversarial Framework. Using fully convolutional networks has the great benefit of not having to stick to a precise input resolution for frames and most importantly allows us to train the network over smaller frame crops and larger batches, speeding up the training. Considering the fact that the noise process induced by compression is local, our strategy does not compromise performance.

We apply adversarial training [12] which has recently shown remarkable performances in image processing and image generation tasks [4, 9, 17]. GAN training consist of the optimization of two networks named *generator* (G) and *discriminator* (D) where the generator is fed some noisy input and has the goal to create "fake" images able to induce the discriminator in mistakes. On the other hand the discriminator optimizes a classification loss rewarding solutions that correctly distinguish *fake* from real images. In our case we are not aiming at generating novel unseen images sampled from a distribution but our task regards the enhancement of a corrupted image. Interestingly such task can be tackled with GANs by conditioning the training. Our end goal is to obtain a $G(\cdot)$ function able to process compressed frames and remove artifacts. In our conditional GAN we provide to the discriminator positive examples $I_t|I_t^C$ and negative examples $I_t^R|I_t^C$, where $\cdot|\cdot$ indicates channel-wise concatenation. This means that, in case of samples of size $N \times N \times Ch$, the discriminator receives a sample with dimensions $N \times N \times 2 \cdot Ch$.

3.1 Generative Network

The architecture of our generator is based on MobileNetV2 [25], which is a very efficient network designed for mobile devices to perform classification tasks. Differently from [9], we replace standard residual blocks with bottleneck depth-separable convolutions blocks, as shown in Table 1, to reduce the overall amount of parameters. We set the expansion factor t to 6 for all the experiments.

Layer	Output
Conv2d 1×1, ReLU6	$m \times n \times t * c$
Dw Conv2d 3×3, ReLU6	$m \times n \times t * c$
Conv2d 1×1	$m \times n \times c$

Table 1: Bottleneck residual block used in our generator network.

After a first standard convolutional layer, feature maps are halved twice with strided convolutions and then we apply a chain of B bottleneck residual blocks. The number of convolution filters doubles each time the feature map dimensions are halved. We use two combinations of nearest-neighbour up-sampling and standard convolution layer to restore the original dimensions of feature maps. Finally, we generate the RGB image with a 1×1 convolution followed by a *tanh* activation, to keep the output values between the $[-1, 1]$ range. In all our trained models we employed Batch Normalization to stabilize the training process. Table 2 reports the number of filters, blocks and weights of the GAN used in a previous work [9], and two variations of the proposed network, called "Fast" and "Very Fast" since they are designed to attain real-time performance. It can be observed that the new GAN architectures have much smaller number of parameters, resulting in reduced computational costs, that allow to reach the required real-time performance.

Model	# Filters	# Blocks	# Params
Galteri *et al.* [9]	64	16	5.1M
Our Fast	32	12	1.8M
Our Very Fast	8	16	145k

Table 2: Parameters of the different GANs used. Compared to the previous work [9], our new "Fast" and "Very Fast" networks have much smaller number of parameters, resulting in improved computation time.

3.2 Discriminative Network

The structure of the discriminator network comprises mostly convolutional layers followed by LeakyReLU activation, with a final dense layer and a sigmoid activation. Since the complexity of this network does not affect the execution time during test phase, we have chosen for all our trainings a discriminator with a very large number of parameters, thus increasing its ability to discriminate fake patches from real ones. As in [9, 10], sub-patches are fed to this network rather than whole images, because image compression operates at sub-patch level and those artifacts we aim to remove are generated inside them. The set of weights ψ of the D network are learned by minimizing:

$$\begin{aligned} l_d = &- \log\left(D_\psi\left(I|I^C\right)\right) \\ &- \log\left(1 - D_\psi\left(I^R|I^C\right)\right) \end{aligned} \tag{2}$$

where $D(x)$ is taken from of the sigmoid activation of the discriminator network, with x indicating the concatenation on channels axis between the distorted input I^C and the correspondent real image I or the synthetic one I^R.

3.3 Content Losses

Here we describe the content losses used in combination with the adversarial loss for the generator. Content losses have the goal to limit the set of distributions to be modeled via the adversarial learning process inducing the generator to produce consistent image enhancement behavior.

Pixel-wise MSE Loss Mean Squared Error loss (MSE) is defined as:

$$l_{MSE} = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \left(I_{x,y} - I^R_{x,y}\right)^2 . \tag{3}$$

This loss is commonly used in image reconstruction and restoration tasks [6, 21, 28]. l_{MSE} recovers low frequency details from a compressed image, but the drawback is that high frequency details are suppressed.

Perceptual Loss Many contributions on image enhancement, restoration and super-resolution have employed the perceptual loss to optimize the network in a feature space rather than the pixel space [2, 7, 11, 15]. We used such loss in our adversarial training to encourage reconstructed images and target ones to have similar feature representations. The similarity measure between two images is obtained by projecting I and I^R on a feature space of a pre-trained network, hence extracting some meaningful feature maps. The perceptual loss is the Euclidean distance between the extracted feature representations:

$$l_P = \frac{1}{W_f H_f} \sum_{x=1}^{W_f} \sum_{y=1}^{H_f} \left(\phi_j \left(I \right)_{x,y} - \phi_j \left(I^R \right)_{x,y} \right)^2 \tag{4}$$

where $\phi_j(I)$ indicates the activations of some j-th layer of the pre-trained network for an input image I, and W_f and H_f are respectively the width and the height of the feature maps. In this work we have chosen the VGG-19 model [26] as feature extractor, using the outputs of the `pool4` layer of this network.

Adversarial Loss The total loss of our generator is a weighted combination of the aforementioned losses:

$$l_{AR} = l_{MSE} + \lambda_1 l_P + \lambda_2 l_{adv}. \tag{5}$$

where l_{adv} is the standard adversarial loss:

$$l_{adv} = -\log \left(D_\psi \left(I^R | I^C \right) \right) \tag{6}$$

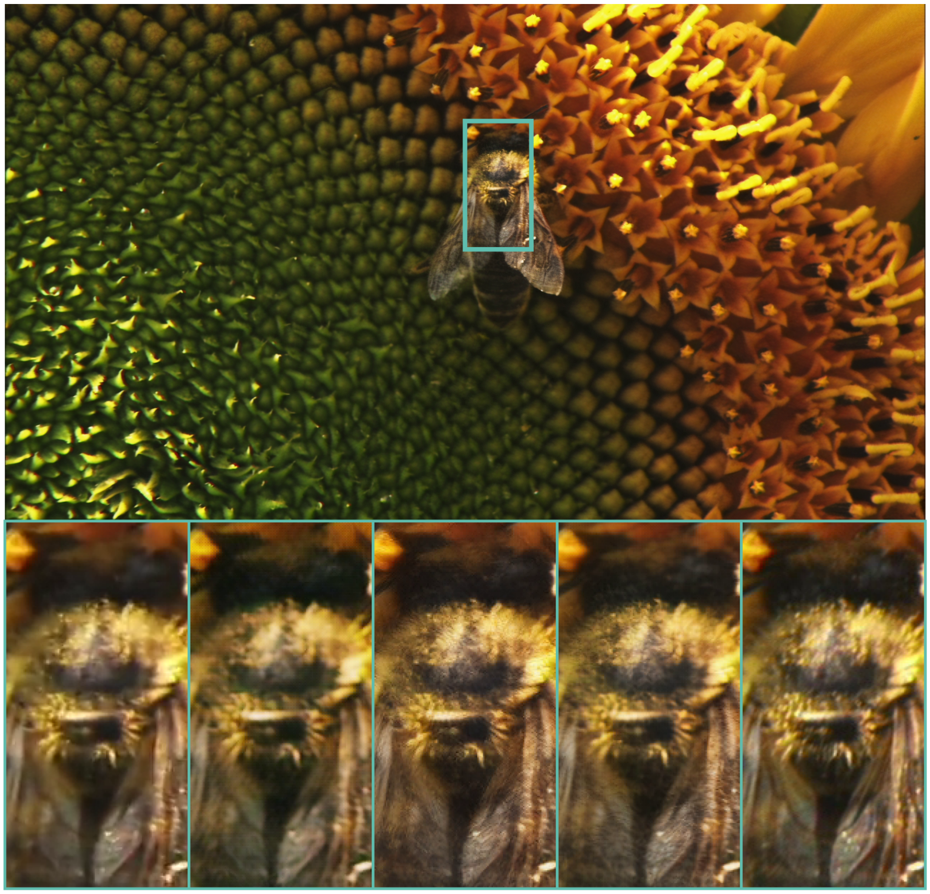
that rewards solutions that are able to “fool” the discriminator.

4 Experimental Results

We test our novel architectures with three popular no reference metrics. No reference (NR) image quality assessment (IQA) is the task of providing a score for an image, which has been possibly distorted by an unknown process, without having access to the original image. These metrics are designed to identify and quantify the presence of different types of artifacts that may be present in the image being analyzed, like blurriness, blocking, lack of contrast or saturation, etc.

All models are trained on DIV2K dataset [1]. DIV2K training set comprises 800 high resolution uncompressed images, which we compress using H.264 to generate degraded frames. As an augmentation strategy, considering the small size of DIV2K, we resize images at 256, 384 and 512 on their shorter side and then we randomly crop a patch of 224×224 pixels with random mirror flipping. This procedures allows to increase dataset size and increase diversity of pattern scales.

Looking at qualitative examples in Fig. 2 and Fig. 3, it can be seen that the quality of imagery is largely improved by our networks in comparison with the source compressed frame; in particular finer details of the wings and hair of the bee and texture of water and



(a) (b) (c) (d) (e)

Fig. 2: Qualitative comparison of (leftmost) compressed frame with H.264 (CRF 28), (b-d) our Very Fast, Fast and Galteri *et al.* [9] networks with (e) uncompressed frame. Large frame obtained by our Fast network. Note the fine details of the wings and hairs of the bee obtained by the GAN based approaches, compared to the standard compressed version.

feathers, as well as reduced ringing and blocking in the body of the duck. As also shown in [9, 10, 33], GAN image enhancement can lead to low performance in full reference metrics due to the fact that the overall pictures is improved by semantically consistent textures which, pixel-wise, may differ from the original uncompressed image.

As a test dataset we pick the following videos: *Mobile Calendar*, *Park Run*, *Shields*, *River Bed*, *Sunflower*, *Rush Hour*, *Tractor Pedestrian Area*,*Blue Sky* and *Station* from the Derf collection[1], since it consists of high resolution uncompressed videos allowing us to test the effect of compression on frames that have not undergone any corruption

[1] https://media.xiph.org/video/derf/

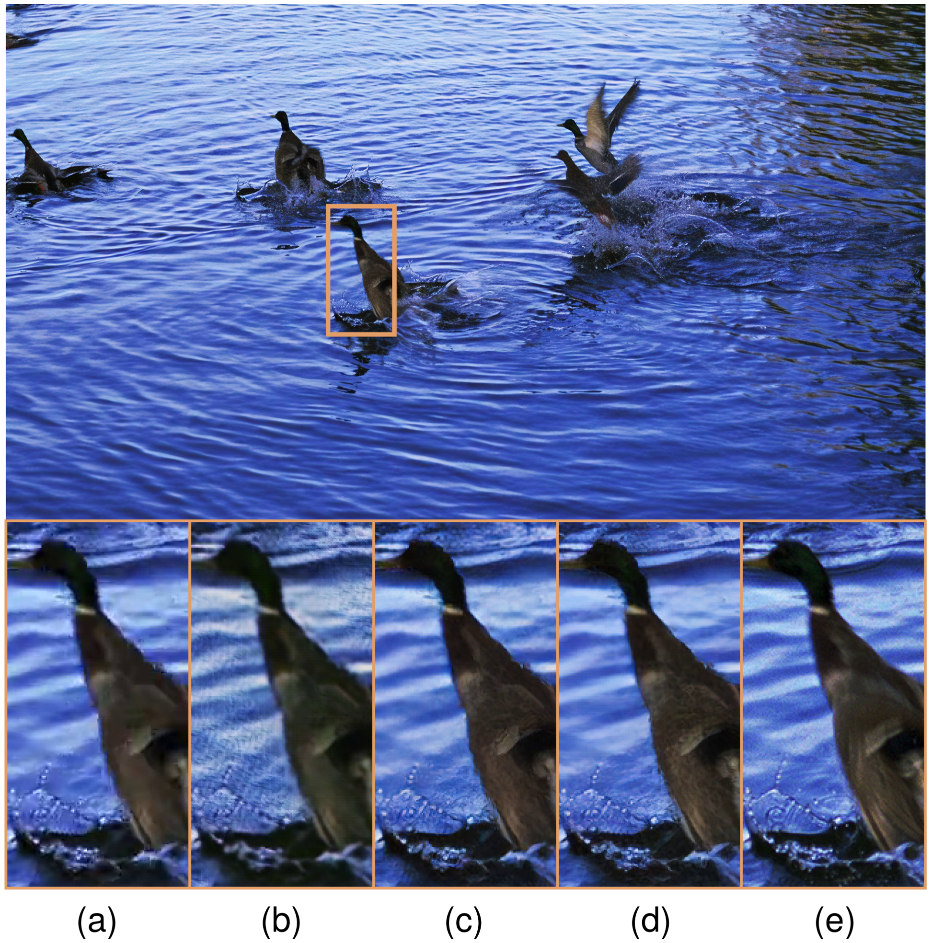


Fig. 3: Qualitative comparison of (a) compressed frame with H.264 (CRF 28), (b-d) our Very Fast, Fast and Galteri *et al.* [9] networks with (e) uncompressed frame. Large frame obtained by our Fast network. Note the fine details of the texture of the water and feathers of the duck obtained by the GAN based approaches, compared to the standard compressed version. Also ringing and blocking artifact in the body of the duck, that are present in the standard compressed version are eliminated.

due to the compression process. We compress video aggressively, setting the constant rate factor (CRF) to 28; in comparison to a more high quality CRF setting (e.g. 10) this results in a $\sim 10\times$–$20\times$ reduction of video size.

We use two image based no reference metrics (BRISQUE [22] and NIQE [24]) and a video specific no reference metric (VIIDEO [23]). We use different metrics since typically NR IQA methods are defined to handle specific types of artifacts. We report results in Tab. 3, for all metrics a lower score corresponds to a better image quality.

For all the image quality enhancement approaches we report their time performance in terms of FPS, computed on videos at 720p resolution.

Interestingly our networks, even our "Very Fast", consistently improve image quality metrics. We run all experiments using a NVIDIA Titan Xp GPU card and our TensorFlow implementation. No further optimizations have been made such as using quantized networks or employing faster inference engines such as Caffe or TensorRT. We use as baseline the GAN presented in [9], which runs at 4 FPS. Interestingly our "Fast" network, running at 20 FPS, obtains comparable results. Our "Very Fast" network, running at 42 FPS is able to improve the compressed frames according to all metrics but does not reach the quality of original frames. When processing HD frames (i.e.1080p resolution) the GAN used in [9] reaches 2 FPS, while our "Very Fast" still obtains real-time performance at 20 FPS.

It can be noted that for NIQE and BRISQUE measures the score measured on enhanced frames for Galteri*et al.* [9] and the Fast network exceeds the one of raw frames. We believe that this must be read with obvious caution, and is likely to be motivated by the fact that these two metrics are assessing frames independently. Looking at the VIIDEO measure all of the proposed networks obtains improve consistently with respect to the compressed frames, but are not better quality than the raw frames.

	VIIDEO [23]	NIQE [24]	BRISQUE [22]	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.* [9]	0.387	**3.594**	17.58	4
Uncompressed	0.276	4.329	23.73	-

Table 3: 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.

5 Conclusion

Many applications based on video streaming impose a strict real-time constraint. Unfortunately existing image quality methods, that improve the quality of reconstruction of compressed frames, are not able to satisfy it. In this paper we move the first steps towards Real-Time GANs for image enhancement. Our Fast network is able to run at 20 FPS with no deterioration on the final results according to three popular image quality measures. Qualitative inspection of our frames confirm quantitative results, showing pleasant highly detailed frames.

Further speed-up can be obtained by exploiting specialized inference engines such as TensorRT and quantizing the models on 8bit. Our current solution do not employ any temporal coherence schema, we expect that adding a loss imposing frame-by-frame coherence may enhance further video quality.

Acknowledgments We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPUs used for this research.

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proc. of IEEE CVPR Workshops (2017)
2. Bruna, J., Sprechmann, P., LeCun, Y.: Super-resolution with deep convolutional sufficient statistics. CoRR **abs/1511.05666** (2015)
3. Cavigelli, L., Hager, P., Benini, L.: CAS-CNN: A deep convolutional neural network for image compression artifact suppression. In: Proc. of IJCNN (2017)
4. Chu, M., Xie, Y., Leal-Taixé, L., Thuerey, N.: Temporally coherent GANs for video super-resolution (tecogan). arXiv preprint arXiv:1811.09393 (2018)
5. Dar, Y., Bruckstein, A.M., Elad, M., Giryes, R.: Postprocessing of compressed images via sequential denoising. IEEE Transactions on Image Processing **25**(7), 3044–3058 (July 2016)
6. Dong, C., Deng, Y., Change Loy, C., Tang, X.: Compression artifacts reduction by a deep convolutional network. In: Proc. of ICCV (2015)
7. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Proc. of NIPS (2016)
8. Foi, A., Katkovnik, V., Egiazarian, K.: 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)
9. Galteri, L., Seidenari, L., Bertini, M., Del Bimbo, A.: Deep universal generative adversarial compression artifact removal. IEEE Transactions on Multimedia pp. 1–1 (2019)
10. Galteri, L., Seidenari, L., Bertini, M., Del Bimbo, A.: Deep generative adversarial compression artifact removal. In: Proc. of ICCV (2017)
11. Gatys, L.A., Ecker, A.S., Bethge, M.: Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. CoRR **abs/1505.07376** (2015)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc. of NIPS (2014)
13. He, X., Hu, Q., Zhang, X., Zhang, C., Lin, W., Han, X.: Enhancing HEVC compressed videos with a partition-masked convolutional neural network. In: Proc. of ICIP (2018)
14. Jakhetiya, V., Lin, W., Jaiswal, S.P., Guntuku, S.C., Au, O.C.: Maximum a posterior and perceptually motivated reconstruction algorithm: A generic framework. IEEE Transactions on Multimedia **19**(1), 93–106 (2017)
15. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proc. of ECCV (2016)
16. Kang, L.W., Hsu, C.C., Zhuang, B., Lin, C.W., Yeh, C.H.: Learning-based joint super-resolution and deblocking for a highly compressed image. IEEE Transactions on Multimedia **17**(7), 921–934 (2015)
17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948 (2018)
18. Li, T., He, X., Qing, L., Teng, Q., Chen, H.: An iterative framework of cascaded deblocking and super-resolution for compressed images. IEEE Transactions on Multimedia (2017)
19. Li, Y., Guo, F., Tan, R.T., Brown, M.S.: A contrast enhancement framework with JPEG artifacts suppression. In: Proc. of ECCV (2014)
20. List, P., Joch, A., Lainema, J., Bjontegaard, G., Karczewicz, M.: Adaptive deblocking filter. IEEE Transactions on Circuits and Systems for Video Technology **13**(7), 614–619 (2003)

21. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Proc. of NIPS (2016)
22. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. IEEE Transactions on Image Processing **21**(12), 4695–4708 (Dec 2012)
23. Mittal, A., Saad, M.A., Bovik, A.C.: A completely blind video integrity oracle. IEEE Transactions on Image Processing **25**(1), 289–300 (2016)
24. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a completely blind image quality analyzer. IEEE Signal Processing Letters **20**(3), 209–212 (2013)
25. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: Proc. of CVPR (June 2018)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proc. of ICLR (2015)
27. Stockhammer, T.: Dynamic adaptive streaming over http–: standards and design principles. In: Proc. of ACM MMSys. pp. 133–144. ACM (2011)
28. Svoboda, P., Hradis, M., Barina, D., Zemcik, P.: Compression artifacts removal using convolutional neural networks. arXiv preprint arXiv:1605.00366 (2016)
29. Wang, Z., Liu, D., Chang, S., Ling, Q., Yang, Y., Huang, T.S.: D3: Deep dual-domain based fast restoration of JPEG-compressed images. In: Proc. of CVPR (2016)
30. Wong, T.S., Bouman, C.A., Pollak, I., Fan, Z.: A document image model and estimation algorithm for optimized JPEG decompression. IEEE Transactions on Image Processing **18**(11), 2518–2535 (2009)
31. Yang, J.X., Wu, H.R.: 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)
32. Yang, S., Kittitornkun, S., Hu, Y.H., Nguyen, T.Q., Tull, D.L.: Blocking artifact free inverse discrete cosine transform. In: Proc. of ICIP (2000)
33. Yoo, J., Lee, S.h., Kwak, N.: Image restoration by estimating frequency distribution of local patches. In: Proc. of CVPR (2018)
34. Zhang, J., Xiong, R., Zhao, C., Zhang, Y., Ma, S., Gao, W.: CONCOLOR: Constrained non-convex low-rank model for image deblocking. IEEE Transactions on Image Processing **25**(3), 1246–1259 (March 2016)
35. Zhang, X., Xiong, R., Fan, X., Ma, S., Gao, W.: Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. IEEE Transactions on Image Processing **22**(12), 4613–4626 (2013)