

Deep Universal Generative Adversarial Compression Artifact Removal

Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo, *Member, IEEE*

Abstract—Image compression is a need that arises in many circumstances. Unfortunately, whenever a lossy compression algorithm is used, artifacts will manifest. Image artifacts, caused by compression tend to eliminate higher frequency details and in certain cases may add noise or small image structures. There are two main drawbacks of this phenomenon. First, images appear much less pleasant to the human eye. Second, computer vision algorithms such as object detectors may be hindered and their performance reduced. Removing such artifacts means recovering the original image from a perturbed version of it. This means that one ideally should invert the compression process through a complicated non-linear image transformation. We propose an image transformation approach based on a feed-forward fully convolutional residual network model. 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 GAN is able to produce images with more photorealistic details than SSIM based networks. We describe a novel training procedure based on sub-patches and devise a novel testing protocol to evaluate restored images quantitatively. We show that our approach can be used as a pre-processing step for different computer vision tasks in case images are degraded by compression to a point that state-of-the art algorithms fail. In this case, our GAN-based approach obtains better performance than MSE or SSIM trained networks. Differently from previously proposed approaches we are able to remove artifacts generated at any QF by inferring the image quality directly from data.

Index Terms—Image Compression, Image Restoration, Object Detection

I. INTRODUCTION

Every day billions of images are shared on the web, and many more are produced and kept on private systems as mobile phones, cameras and surveillance systems. To practically store and transmit these images it is necessary to compress them, in order to reduce bandwidth and storage requirements. Apart from a few cases where compression has to be lossless, e.g. medical imaging or technical drawings, the algorithms used are lossy, i.e. they result in a more or less strong loss of content fidelity with respect to the original image data, to achieve a better compression ratio. A typical use case in which a high compression is desirable is that of web images, in which image files must be kept small to reduce web page 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 tasks such as entertainment video streaming, like Netflix, there is need to reduce as much as possible the required bandwidth,

to avoid network congestions and to reduce costs. Since user experience is also affected by image quality, compression algorithms are designed to reduce perceptual quality loss, according to some model of the human visual system. In fact, when compressing images several artifacts appear as shown in Fig. 1. These artifacts are due to the different types of lossy compressions used. Considering JPEG, the most common algorithm used nowadays, these artifacts are due to the chroma subsampling (i.e. dropping some color information of the original image) and the quantization of the DCT coefficients; these effects can be observed also in MPEG compressed videos, that is basically based the same schema with the addition of motion compensation and coding.

In the past, compression artifact removal has been addressed mainly without learning from large dataset the denoising function. This can be done optimizing DCT coefficients [54] or by regularizing image patches based on adaptive distribution modeling. The majority of existing work is not using any learning and is not considering the use of deep convolutional neural networks (CNN). CNNs have been proposed for reducing artifacts in two works [9], [43] and for image denoising [53]. Nonetheless this approach has been used fruitfully in super-resolution[26], that is the task of generating larger images by adding missing details to down-sampled sources.

In this work we propose a solution to artifact removal based on convolutional neural networks trained on large sets of patches compressed at different qualities. Our approach can work as a post-processing on decompressed images and can therefore be applied on many lossy compression algorithms such as JPEG, JPEG2000, WebP and the intra-frame coding of H.264/AVC and H.265/HEVC.

One of the main advantages of 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 usually hardware based compression pipeline. It is also more common that a streaming signal changes in quality over time to cope with bandwidth availability. This would not be true in case we rely on super-resolution which would require image sub-sampling also on the coding end.

To assess the performance of the artifact removal process, and the quality of restored images, there is need to assess both subjective and objective evaluations. The former are needed since most of the time a human will be the ultimate consumer of the compressed media. The latter are important since obtaining subjective evaluations is slow and costly; to this end several objective metrics have been proposed to predict perceived visual quality automatically. Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) are the most widely used objective image quality/distortion metrics.

L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo are with the Media Integration and Communication Center (MICC), Università degli Studi di Firenze, Viale Morgagni 65, 50134 - Firenze, Italy.

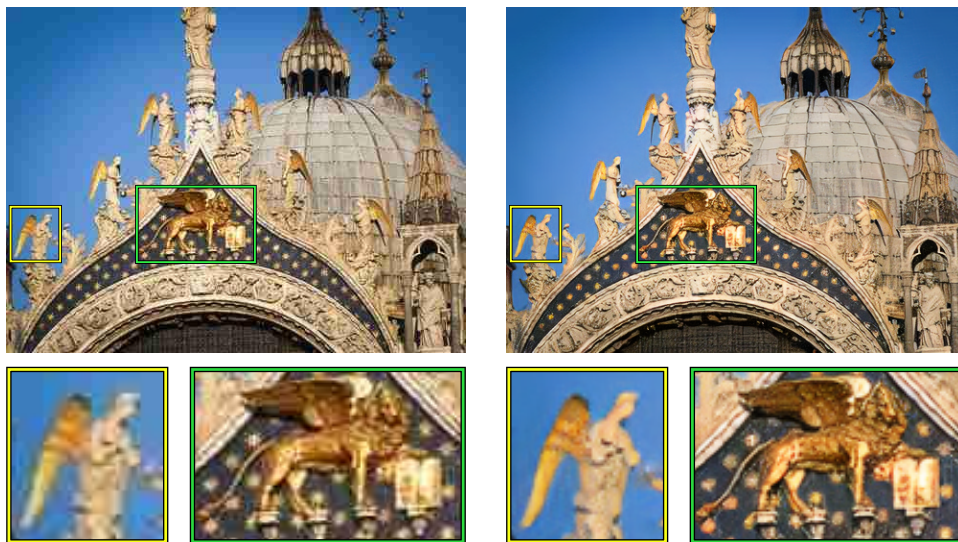


Fig. 1: Left: An image compressed using JPEG. Two highly degraded regions are highlighted. Right: Output of our reconstruction method. Both regions appear sharper and with far less artifacts. Best viewed in color on computer screen.

However, they have been criticized since they are not consistent with perceived quality measurement [44]. Considering that the human visual system is highly adapted for extracting structural information from a scene, a framework for quality assessment based on the degradation of structural information, called Structural Similarity index (SSIM), has been introduced in [45]. Finally, we can expect that more and more images will be processed by computer vision systems that automatically analyze media content, e.g. to interpret it to perform some task. To consider also this scenario we have to assess the performance of computer vision algorithms when processing reconstructed images.

In this work we show that it is possible to train CNNs to remove compression artifacts even from highly degraded images. Our network can be trained optimizing directly the SSIM on output images. This approach leads to state-of-the-art results. However, it can be shown that SSIM is yet a too simplistic model assess quality according to the complex human visual system. We show that Generative Adversarial Models, learning the conditional distribution of compressed and uncompressed images, lead to better reconstruction. We provide a system capable of understanding image quality automatically which can therefore reconstruct images at any level of compression.

We assess the performance of our approach using both subjective and objective assessments. We design a novel experimental protocol to assess the quality of reconstructed images based on the evaluation of a semantic task on restored images. GAN reconstruction provides higher fidelity images according to human viewers and higher performance in object detection.

II. RELATED WORK

There is a vast literature of image restoration, targeting image compression artifacts. The majority of approaches is processing based [8], [12], [20], [27], [28], [48], [49], [52],

[54] while few methods are learning based [22], [9], [31], [43], [46]. In the following we will review both kind of methods. We will also cover other works solving different image transformation tasks which are related to our problem. Finally we will state our contributions in relation to existing state of the art.

A. Processing Based Methods

This class of methods typically relies on information in the DCT domain. Foi *et al.* [12] developed the SA-DCT method, based on the use of clipped or attenuated DCT coefficients to reconstruct a local estimation of the image signal within an adaptive shape support. Yang *et al.* [49], applied a DCT-based lapped transform directly in the DCT domain, to remove the artifacts produced by quantization. Zhang *et al.* [54], proposed to fuse two predictions for estimating DCT coefficients of each image block: the first prediction is based on quantized values of coefficients and the second is computed from nonlocal blocks coefficients as a weighted average. Li *et al.* [28] have proposed to eliminate the artifacts due to contrast enhancement, through decomposition of the image in structure and texture components, and then eliminating the artifacts that are in the texture component. Chang *et al.* [5] have proposed to obtain a sparse representation over a learned dictionary from a set of training images, and then use it to remove the block artifacts in JPEG-compressed images. More recently, Dar *et al.* [8] have proposed to reduce artifacts through a regularized restoration of the original signal. The procedure is formulated as a regularized inverse-problem for estimating the original signal from its reconstructed form; to obtain a tractable formulation the nonlinear compression-decompression process is approximated by a linear operator. Finally, Li *et al.* [27] have used an iterative approach to address blocking artifacts; this method can also perform super-resolution.

The main issue of these methods is that the reconstructed image is typically over-smooth. In fact, it is hardly possible to add consistent details at higher frequencies without any semantic cue of the content of the image.

B. Learning Based Methods

Following the success of deep convolutional neural networks (DCNN), a learning driven paradigm has recently emerged in the artifact removal literature. The main idea of this strategy is to learn a function to perform an image transformation from a degraded input image to a restored output. Labeled data can be easily obtained by generating degraded versions of images which are used as samples for which the ground truth or target is the original image. Learning based methods have the advantage that they estimate very accurately the image manifold, thanks to the large amount of data that they ingest during training. Moreover, such manifold can also be made aware of image semantics and is not just relying on local properties or DCT coefficient statistics.

Kang *et al.* [22] 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. Following their previous work on super-resolution CNN (SRCNN), Dong *et al.* [9] propose an artifact reduction CNN (AR-CNN) which shares a common structure with SRCNN: a feature extraction layer, a feature enhancement layer, and a non-linear mapping and a reconstruction layer. This structure is designed following sparse coding pipelines. Svoboda *et al.* [43] obtain improved results in image restoration by learning a feed-forward CNN in which, differently from [9], the layers have no specific functions; to obtain better reconstruction quality the authors combine residual learning, skip architecture and symmetric weight initialization. Cavigelli *et al.* [4] use a 12-layers CNN with hierarchical skip connections and a multi-scale loss function to suppress JPEG compression artifacts, proposing an architecture that is able to shorten the paths from input to output, so to ease the training. Yoo *et al.* [51] aim at restoring high-frequency details in JPEG compressed images employing an encoder-decoder architecture, driven by a local frequency classifier to restore compressed images; cross-entropy is used to train the classifier, and MSE loss is used for encoder-decoder. He *et al.* [18] have developed a method, tightly bound to HEVC coding, to improve frame appearance; it smartly exploits coding unit partitioning to learn a two-stream CNN that receives the decoded frame, and then combines it with a mask computed from the partition data.

A few recent approaches tackle the problem from a different angle by designing the image coding algorithm based on learning a latent representation[2], [39]. The main drawback of such approaches is that they can not be applied to existing low quality images and they require that both parties involved in the image transmission adopt the learning based codec. In our case we only act on the receiving side of the communication party, therefore our method is more flexible and applicable to existing compressed data.

C. Other Image Transformation Tasks

Other image transformation problems, such as image super-resolution [3], [26], [21], [7], [6], [25], style-transfer [15], [21] and image de-noising [53] have been targeted by approaches close to ours. Zhang *et al.* [53] have recently addressed the problem of image denoising, proposing a convolutional neural networks to eliminate unknown level Gaussian noise and showing that single residual units of the network combined with batch normalization are beneficial. The proposed network obtains promising results also on other tasks such as super resolution and JPEG deblocking. Style transfer is the process of altering an image so that its semantic content remains the same but its style is altered, transferring it from another image. Gatys *et al.* [15] have shown that optimizing a loss accounting for style and content similarity it is possible to perform this task. Similarly, Johnson *et al.* [21] propose a generative approach to solve style transfer, building on the method of [15]. The improvement in terms of performance, with respect to [15], is due to the fact that optimization is performed beforehand, for each style; moreover, it is possible to apply the transformation in real-time. Adding a slight variation on the learning procedure they are able to perform also super-resolution. Regarding super-resolution, Kim *et al.* [23] propose to use a deeper architecture (VGG, [42]) trained on residual images; in order to speed-up learning they apply gradient clipping. Bruna *et al.* [3] addressed super-resolution using a CNN to learn sufficient statistics for the high-frequency component. Ledig *et al.* [26] used a deep residual convolutional generator network, trained in an adversarial fashion. Dahl *et al.* [7] propose to use a PixelCNN architecture and apply it to magnification of 8×8 pixel images obtaining better quality results compared to L2 regression according to human evaluators.

D. Contribution

In this paper we make several contributions to the problem of image enhancement. Existing learning based methods [9], [4], [51], [13] are trained to remove artifacts generated by some encoder knowing the parameters in advance. Considering that this is an unrealistic setting, we address this issue proposing an ensemble of Generative Adversarial Networks [16] driven by a quality predictor. We show experimentally that quality can be predicted effectively and that we can enhance images without knowing the encoding parameters in advance.

Our model is fully convolutional as the one proposed by Svoboda *et al.* [43] and can therefore process images at any input resolution. Differently from [43] we use a deep residual architecture[17] and use Generative Adversarial Networks. We show that our model can be trained with direct supervision with a MSE loss as in [43] or with a better SSIM based loss. Nonetheless such training procedure leads to overly smoothed images as also happens in super-resolution.

Exploiting GANs instead, considering their ability to model complex multi-modal distributions, we are able to obtain sharper and more realistic images. To the best of our knowledge, this is the first work exploiting multiple GANs to recover from compression artifacts generated by an encoder at

a unknown quality. We train conditional GANs [33], to better capture the image transformation task. A relevant novelty of our work is the idea of learning the discriminator over sub-patches of a single generated patch to reduce high frequency noise, such as mosquito noise which is hard to remove using a full-patch discriminator.

Another major contribution of this work is the evaluation methodology. Instead of focusing on signal based metrics we exploit well defined semantic tasks and evaluate its performance on reconstructed images. Specifically, we evaluate two tasks: object detection and object mask proposal generation.

We improved our previous work [13] proposing an ensemble of GAN models, each specialized on a single QF; we drive the ensemble with our QF prediction framework which is described in Sect. IV. Newer and more up to date experiments are provided in Sect. V, including detection and segmentation tests on MS-COCO and additional comparisons on PASCAL VOC. Moreover a full in-depth evaluation in realistic settings, i.e. when image encoder parameters are not known in advance, is performed in Sect. V-D. Interesting insights on our method can be gained following our novel evaluation approach; specifically, in Sect. V-C3 we analyze the correlation between the degradation of intermediate feature maps of object detectors and the resulting performance drop.

III. METHODOLOGY

The goal of compression artifact removal is to obtain a reconstructed output image I^R from a compressed input image I^C . In this scenario, $I^C = A(I)$ is the output image of a compression algorithm A and I is an uncompressed input image. Different A algorithms will produce different I^C images, with different compression artifacts. Many image and video compression algorithms (e.g. JPEG, JPEG2000, WebP, H.264/AVC, H.265/HEVC) work in the YCrCb color space, separating luminance from chrominance information. This allows a better de-correlation of color components leading to a more efficient compression; it also permits a first step of lossy compression sub-sampling chrominance, considering the reduced sensitivity of the human visual system to its variations.

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

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

$$I^C = A(I, QF) \in [0, 255]^{W \times H \times C} \quad (1)$$

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

$$G(I^C) = I^R \approx I \quad (2)$$

Each generator can in principle be trained with images obtained from different QFs. In practice we show, in Sect. IV, that single QF generators perform better and can be driven by a QF predictor.

To this end, we train a convolutional neural network $G(I^C; \theta_g)$ 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:

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

The elimination of compression artifacts is a task that belongs to the class of image transformation problem, that comprises other tasks such as super-resolution and style-transfer. This category of tasks is conveniently addressed using generative approaches, i.e. learning a fully convolutional neural network (FCN) [30] that given a certain input image is able to output an improved version of it. A reason to use FCN architectures in image processing is that they are extremely convenient to perform local non-linear image transformations, and can process images of any size. Interestingly, we take advantage of such property to speed up the training. Indeed, the artifacts we are interested in removing appear at scales close to the block size. For this reason we can learn models on smaller patches using larger batches.

We propose a fully convolutional architecture that can be either optimized with direct supervision or combined in a generative adversarial framework with a novel discriminator. Details of the proposed networks are presented in the following, together with the devised loss functions.

A. Generative Network

In this work we use a deep residual generative network, composed only by blocks of convolutional layers with non-linear LeakyReLU activations. Our generator is inspired by [17]. We use layers with 64 convolution kernels with a 3×3 support, followed by LeakyReLU activations. After a first convolutional layer, we apply a layer with stride two to half the size of feature maps. Then we apply 15 residual blocks using a 1 pixel padding after every convolution with replication strategy to mitigate border effects. A nearest-neighbour upsampling layer is used to obtain feature maps at the original size[35]. Considering that upsampling may lead to artifacts we apply another stride one convolutional layer. Finally, to generate the image we use single kernel convolutional layer with a \tanh activation. This produces output tensors with values in $[-1, 1]$, which are therefore comparable to the rescaled image input. Adding batch normalization helps training of the GAN, resulting in a moderately improved performance, as shown in Sect. V-C1.

B. Loss Functions for Direct Supervision

In this sub-section we discuss how to learn a generative network with direct supervision, i.e. computing the loss as a function of the reconstructed image I^R and of the original

uncompressed input image I . Classical backpropagation is used to update the network weights.

1) *Pixel-wise MSE Loss*: As a baseline we use the Mean Squared Error loss (MSE):

$$l_{MSE} = \frac{1}{WH} \sum_{x=1}^W \sum_{y=1}^H (I_{x,y} - I_{x,y}^R)^2. \quad (4)$$

This loss is commonly used in image reconstruction and restoration tasks [9], [43], [31]. It has been shown that l_{MSE} is effective to recover the low frequency details from a compressed image, but the drawback is that high frequency details are suppressed.

2) *SSIM Loss*: The Structural Similarity (SSIM) [45] has been successfully proposed as an alternative to MSE and Peak Signal-to-Noise Ratio (PSNR) image similarity measures, because both these measures have shown to be inconsistent with the human visual perception of image similarity.

The formula to compute the SSIM of the uncompressed image I and the reconstructed image I^R is:

$$SSIM(I, I^R) = \frac{(2\mu_I\mu_{I^R} + C_1)(2\sigma_{II^R} + C_2)}{(\mu_I^2 + \mu_{I^R}^2 + C_1)(\sigma_I^2 + \sigma_{I^R}^2 + C_2)} \quad (5)$$

Considering that the SSIM function is fully differentiable a loss can be defined as:

$$l_{SSIM} = -\frac{1}{WH} \sum_{x=1}^W \sum_{y=1}^H SSIM(I_{x,y}, I_{x,y}^R) \quad (6)$$

The network can then be trained minimizing Eq. 6, which means maximizing the structural similarity score computed on uncompressed and reconstructed image pairs.

C. Generative Adversarial Artifact Removal

The network defined by the architecture described in Sect. III-A can be coupled with a discriminator and used as generator to obtain a generative adversarial framework. The recent approach of adversarial training [16] has shown remarkable performances in the generation of photo-realistic images and in super-resolution tasks [26]. In this approach, the generator network G is encouraged to produce solutions that lay on the manifold of the real data by learning how to fool a discriminative network D . On the other hand, the discriminator is trained to distinguish reconstructed patches I^R from the real ones I . In particular, we use a conditional generative approach, i.e. we provide as input to the generative network both positive examples $I|I^C$ and negative examples $I^R|I^C$, where $\cdot|$ indicates channel-wise concatenation. For samples of size $N \times N \times C$ we discriminate samples of size $N \times N \times 2C$.

1) *Discriminative Network*: The architecture of the discriminator is based on a series of convolutional layers without padding and with single-pixel stride followed by LeakyReLU activations. The number of filters is doubled every two layers, with the exception of the last one. There are no fully connected layers. The size of the feature map is decreased solely because of the effect of convolutions reaching unitary dimension in the

last layer, in which the activation function used is a sigmoid. A schema of this architecture is shown in Fig.2.

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

$$l_d = -\log(D_\psi(I|I^C)) - \log(1 - D_\psi(I^R|I^C)) \quad (7)$$

As shown Fig. 2, discrimination is performed at the sub-patch level; this is motivated by the fact that compression algorithms decompose images into patches and thus artifacts are typically created within them. To encourage the generation of images with realistic patches, I and I^R are partitioned into P patches of size 16×16 , that are then fed into the discriminator network. In Figure 3 it can be seen the beneficial effect of this approach in the reduction of mosquito noise and ringing artifacts.



Fig. 3: Left: reconstruction without sub-patch strategy. Right: our sub-patch strategy reduces mosquito noise and ringing artifacts.

2) *Perceptual Loss*: Following the contributions of Dosovitskiy and Brox [10], Johnson *et al.* [21], Bruna *et al.* [3] and Gatys *et al.* [14] we use a loss based on perceptual similarity in the adversarial training. 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:

$$l_P = \frac{1}{W_f H_f} \sum_{x=1}^{W_f} \sum_{y=1}^{H_f} (\phi(I)_{x,y} - \phi(I^R)_{x,y})^2 \quad (8)$$

where W_f and H_f are respectively the width and the height of the feature maps. The images reconstructed by the model trained with the perceptual loss are not necessarily accurate according to the pixel-wise distance measure, but on the other hand the output will be more similar from the point of view of feature representation. In this work we compute $\phi(I)$ by extracting the feature maps from a pre-trained VGG-19 model [42], using the second convolution layer before the last max-pooling layer of the network, namely conv5_3.

3) *Adversarial Patch Loss*: We train the generator combining the perceptual loss with the adversarial loss thus obtaining:

$$l_{AR} = l_P + \lambda l_{adv}. \quad (9)$$

Where l_{adv} is the standard adversarial loss:

$$l_{adv} = -\log(D_\psi(I^R|I^C)) \quad (10)$$

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

Discriminator Network

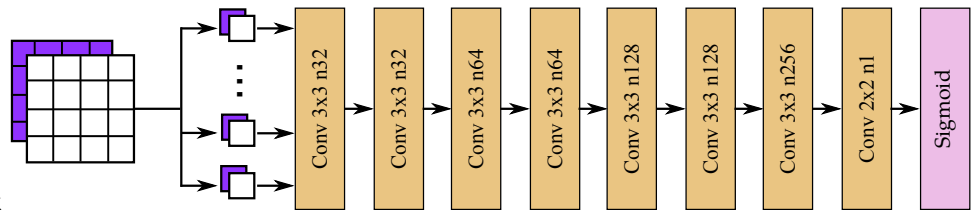


Fig. 2: Architecture of Discriminator Network where n indicates the number of filters for each Convolutional Layer. Sub-patch loss strategy is highlighted: white squares indicate real (I) or reconstructed patches (I^R), while purple ones are their respective compressed versions I^C .

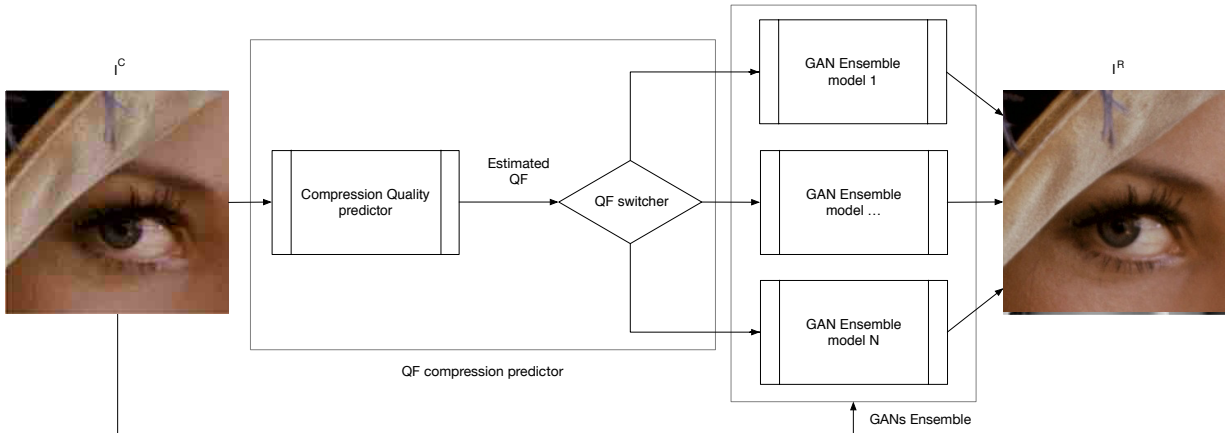


Fig. 4: System schema: the compressed image I^C is analyzed by the compression quality estimator that predicts the QF used to compress the image. This value is used to select an appropriate GAN from the ensemble, to reconstruct the improved image I^R .

IV. UNIVERSAL COMPRESSION ARTIFACT REMOVAL

The quality of an image can not be known in advance. To apply a model in real-world scenarios we can not depend on the prior knowledge of such information. The trivial approach of training a single GAN fed with all QFs is not viable unfortunately; in fact, as shown in Sect. V-D, we observe mode collapse towards higher compression rates. We believe that this effect is due to the fact that images with lower QFs contain more artifacts and generate more signal, thus overcoming the learning of subtle pattern removal that is needed at better qualities.

To cope with this problem, our full solution comprises two modules. The first module predicts, via regression the true QF of an image. This is possible with an extremely high precision. The compression quality estimator is used to drive the image signal to one of the fixed QF trained GANs of our ensemble. A schema of the system is shown in Fig. 4.

A. Quality Agnostic Artifact Removal

Our compression quality predictor consists of a stack of convolutional layers, each one followed by a non-linearity and Batch Normalization, and two Fully Connected layers in the last part. The architecture is shown in detail in Tab. I.

The training set is selected from the DIV2k dataset [1], that contains 800 high definition high resolution raw images. During the training process, we compress the images to a random QF in a 5-95 range and we extract 128×128 patches.

Layer	KernelSize/Stride	OutputSize
Conv11	$3 \times 3/1$	$128 \times 128 \times 64$
Conv12	$3 \times 3/2$	$64 \times 64 \times 64$
Conv21	$3 \times 3/1$	$64 \times 64 \times 128$
Conv22	$3 \times 3/2$	$32 \times 32 \times 128$
Conv31	$3 \times 3/1$	$32 \times 32 \times 256$
Conv32	$3 \times 3/2$	$16 \times 16 \times 256$
Conv41	$3 \times 3/1$	$16 \times 16 \times 512$
Conv42	$3 \times 3/2$	$8 \times 8 \times 512$
FC5	-	1024
FC6	-	1

TABLE I: Network architecture of the proposed QF predictor.

For the optimization, we used a standard MSE loss, computed over predicted and ground truth QF. We train the model as a regressor rather than a classifier since the wrong predictions that are close to the ground truth should not be penalized too much, as the corresponding reconstructions still result acceptable. On the other hand, predictions that are far from the ground truth lead to bad reconstructions, therefore we should penalize them accordingly in the training process.

In the inference phase, we extract 8 random crops of 128×128 from a compressed image, we feed them into the QF predictor and we average the prediction results. We use this prediction to reconstruct the corrupted image with the appropriate model for the input image quality. For this reason, we have trained 6 different generators, each one with fixed QF training images (5,10,20,30,40,60). Depending on the prediction, we give in input the corrupted image to the fixed

QF reconstruction network closer to the QF predictor output.

V. EXPERIMENTS

A. Implementation Details

We trained our reconstruction models with a NVIDIA Maxwell Titan X GPU using MS-COCO [29] as training set, that contains 80 object classes and a total of more than 300K images. In all experiments, we have extracted 16 random 128×128 patches from the training data, with random flipping and rotation data augmentation. All the images have been compressed with the standard MATLAB JPEG compressor at different quality factors to ensure a proper experimental setup both for learning and evaluation. At the training stage, we have used Adam [24] with momentum 0.9 and a learning rate of 10^{-4} for the first 50,000 iterations, decaying to 10^{-5} in the last 50,000. To stabilize the training of the Generative Adversarial framework we have followed the guidelines described in [40], in particular we have performed the one-sided label smoothing for the discriminator training.

B. Comparison with State-of-the-Art

We first report results of our generator network trained without the adversarial approach, evaluating the improvements of the residual architecture and the effects of SSIM and MSE losses in such training. We conducted experiments on two commonly used benchmarks: LIVE1 [41] and the validation set of BSD500 [32] using JPEG as compression. For a fair comparison with the state-of-the-art methods, we adopted the same evaluation procedure of related artifact removal works. To quantify the quality of our results we have evaluated PSNR, PSNR-B [50] and SSIM measures for the JPEG quality factors 10, 20, 30 and 40. The performance of our generator is compared with the standard JPEG compression and three state-of-the-art approaches: SA-DCT [12], AR-CNN from Dong *et al.* [9] and the work described by Svoboda *et al.* [43]. Also the more recent results obtained Cavigelli *et al.* [4] (CAS-CNN), and Yoo *et al.* [51] (ED, CED-EST and CED-GT) are reported, when available.

We report in Table II the results of our approaches on BSD500 and LIVE1 datasets compared to the other state-of-the-art methods for the JPEG artifact removal task. As can be seen, our method outperforms the other approaches for each quality measure, except in two cases: PSNR-B at 10 and 40 where [4] has a slightly better performance. In particular, we have a more remarkable improvement of PSNR and PSNR-B measures for the networks trained with the classic MSE loss, while as expected the SSIM measure improves a lot in every evaluation when the SSIM loss is chosen for training. As a final consideration please note that competing deep models [4], [51] use more parameters, roughly two and three times more than ours.

Furthermore, we report the performance of our generator trained in an adversarial fashion. We can state that we obtain a lower performance than classic approaches from a quality index point of view. However, the GAN reconstructed images are perceptually more convincing for human viewers, as it will be shown in Sect. V-F, in a subjective study. Indeed, it's the

combination of perceptual and adversarial loss that makes the textures of output images much more realistic rather than the smooth patches of the MSE/SSIM based methods that lack of high frequency details, since the latter tend to evaluate better more conservative blurry averages over more photo realistic details, that could be added slightly displaced with respect to their original position, as observed also in super-resolution tasks [7].

QF	Method	LIVE1			BSD500		
		PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
10	JPEG	27.77	25.33	0.791	27.58	24.97	0.769
	SA-DCT [12]	28.65	28.01	0.809	-	-	-
	AR-CNN [9]	29.13	28.74	0.823	28.74	28.38	0.796
	L4 [43]	29.08	28.71	0.824	28.75	28.29	0.800
	CAS-CNN, MS loss[4]	29.36	28.92	0.830	-	-	-
	CAS-CNN, w/ loss FT[4]	29.44	29.19	0.833	-	-	-
	ED [51]	29.40	29.09	0.833	28.96	28.57	0.806
	CED-EST [51]	29.40	29.08	0.832	28.95	28.56	0.805
	CED-GT [51]	26.54	26.51	0.767	26.00	25.97	0.731
	Our MSE	29.47	29.13	0.833	29.05	28.64	0.806
	Our SSIM	28.94	28.46	0.841	28.53	27.97	0.816
	Our GAN	27.65	27.63	0.777	27.31	27.28	0.749
20	JPEG	30.07	27.57	0.868	29.72	26.97	0.852
	SA-DCT [12]	30.81	29.82	0.878	-	-	-
	AR-CNN [9]	31.40	30.69	0.890	30.80	30.08	0.868
	L4 [43]	31.42	30.83	0.890	30.90	30.13	0.871
	L8 [43]	31.51	30.92	0.891	30.99	30.19	0.872
	CAS-CNN, MS loss[4]	31.67	30.84	0.894	-	-	-
	CAS-CNN, w/ loss FT[4]	31.70	30.88	0.895	-	-	-
	ED[51]	31.68	31.14	0.895	31.08	30.33	0.875
	CED-EST [51]	31.65	31.13	0.895	31.04	30.32	0.875
	CED-GT [51]	29.33	29.32	0.854	28.62	28.58	0.825
	Our MSE	31.81	31.29	0.897	31.23	30.49	0.877
	Our SSIM	31.51	30.84	0.901	30.92	30.01	0.883
Our GAN	29.99	29.69	0.864	29.48	29.03	0.841	
30	JPEG	31.41	28.92	0.900	30.98	28.23	0.886
	SA-DCT [12]	32.08	30.92	0.908	-	-	-
	AR-CNN [9]	32.69	32.15	0.917	-	-	-
	Our MSE	33.21	32.51	0.923	32.53	31.57	0.906
	Our SSIM	32.95	32.17	0.926	32.26	31.16	0.911
	Our GAN	31.65	31.38	0.900	31.04	30.57	0.881
	JPEG	32.35	29.96	0.917	31.88	29.14	0.906
40	SA-DCT [12]	32.99	31.79	0.924	-	-	-
	AR-CNN [9]	33.63	33.12	0.931	-	-	-
	CAS-CNN, MS loss[4]	33.98	32.83	0.935	-	-	-
	CAS-CNN, w/ loss FT[4]	34.10	33.68	0.937	-	-	-
	Our MSE	34.17	33.42	0.937	33.45	32.34	0.923
	Our SSIM	33.98	33.07	0.939	33.25	31.94	0.926
	Our GAN	31.64	31.17	0.903	30.98	30.16	0.884

TABLE II: Average PSNR, PSNR-B and SSIM results on BSD500 and LIVE1. Evaluation using luminance. All models are QF-specific.

C. Object Detection

In this experiment we evaluate the object detector performance on images compressed at different QFs. We evaluated the object detector performance for different reconstruction algorithms on PASCAL VOC2007 [11] and MS-COCO [29]. PASCAL VOC2007 is a long standing small scale benchmark for object detection, it comprises 20 classes for a total of roughly 11K images. Regarding MS-COCO [29], we performed detection experiments using the 20,000 images in the *test-dev* subset.

1) *Experiments on VOC2007*: As we can expect, the more an image is degraded by the JPEG compression the lower is the performance of the object detector, especially if the QF parameter is really low. We employ Faster R-CNN [38] as object detector for this experiment and we evaluate its performance on different compression quality versions of PASCAL VOC2007 dataset; we report the results on Tab. IV. To establish an upper bound for this reconstruction task, we report the mean average precision (mAP) on the unaltered dataset. On the other hand, the lower bound is the performance of Faster R-CNN on images JPEG compressed with QF set to

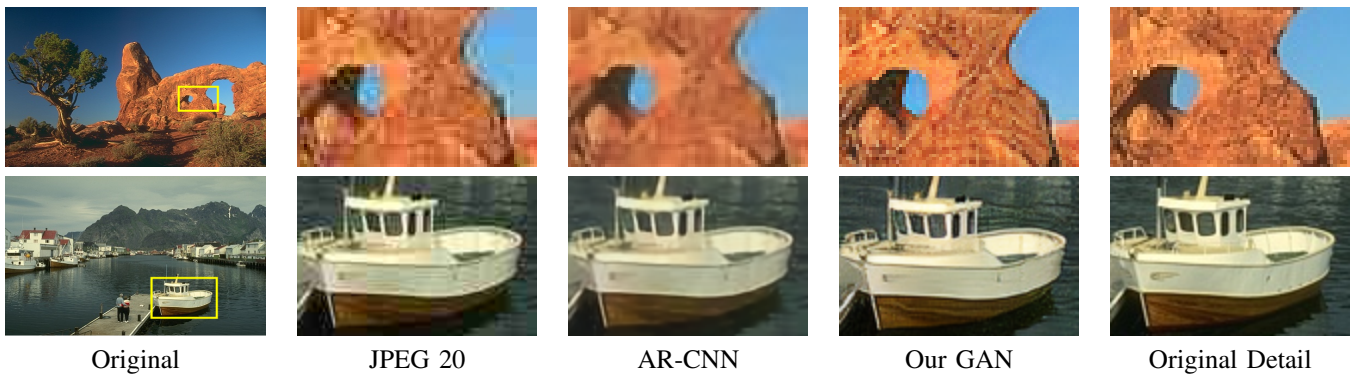


Fig. 5: Qualitative results shown on two complex textured details. JPEG compression introduces severe blocking, ringing and color quantization artifacts. AR-CNN is able to slightly recover but produces a blurry result. Our reconstruction is hardly discernible from the original image.

Codec	Compressed	GAN
WebP	60.1	64.1
JPEG2000	58.7	61.6
BPG	62.3	64.4

TABLE III: Object detection performance (mAP) of our method on other codecs on VOC2007 for similar bitrate.

20 (6,7 \times less bitrate). In this experiment, we evaluate the object detection performance on the different reconstructed versions of the compressed images, comparing AR-CNN [9], our generative MSE and SSIM trained approaches with the GAN. In the first place, we notice that the overall mAP obtained on JPEG compressed images drops down significantly with respect to the upper bound. AR-CNN, MSE and SSIM based generators are able to recover part of the object information, but the improvements compared to the lower bound are not that impressive, as they gain around 2.1, 2.4 and 2.5 points respectively. As we can see in Table IV, our best configuration of the GAN approach is able to restore the degraded images in a much more effective manner yielding the best result increasing the performance by 8.2 points, just 6.0 points less than the upper bound. Compared to our previous result [13], the use of Batch Normalization combined with the QF predictor proposed in Sect. IV (GAN-VGG-BN), improves from 62.3% to 63.1%.

Smaller networks such as AR-CNN [9], are able to achieve reasonable, yet lower, results with respect to our approach. We therefore test a smaller GAN with 7 residual layers to see how much the depth of the network is relevant to obtain quality results. Our GAN recovers 8% mAP points while [9] only adds 3%, when dealing with object detection on compressed images. The smaller network gains 6% mAP point leading to .611, which is still better than [9] but worse than the full network, showing that, as noted for classification tasks [17], [42], network depth matters also for compression artifact removal and image restoration.

Interestingly, our GAN-based approach obtains impressive results on some particular classes, such as *cat* (+16.6), *cow* (+12.5), *dog* (+18.6) and *sheep* (+14.3), i.e. classes with highly articulated objects and where texture is the most informative

cue. In some cases, MSE and SSIM generators are even deteriorating the performance on these categories, as a further confirmation that the absence of higher frequency components alters the recognition capability of an object detector.

Using color gives an obvious advantage in this benchmark, indeed looking at results obtained training the GAN using only luminance (GAN-Y) we lose from 2.5 to 3.3 with respect to GAN-VGG and GAN-VGG-BN. The perceptual loss l_P is relevant in providing sensible semantic cues, this can be seen comparing results with a simpler L1 loss (GAN-L1) which attains much lower performance. Apart from mitigating mosquito noise and ringing, our sub-patch discriminator leads to superior results also in this benchmark. Indeed, training the GAN with a full patch discriminator we obtain 60.5 of mAP while our sub-patch strategy leads to a 62.3 map. The Sub-Patch loss accounts for 1.8% mAP points, highlighting the importance of this novel method.

We analyze the effects of different compression levels in Fig. 6, changing the quality factor of JPEG compressor. As we can see in the figure, GAN approach always outperforms other restoration algorithms; in particular, GAN is able to recover significant details even for very high compression rates, such as QF=10. The gap in performance is reduced when QF raises, e.g. QF=40 (4,3 \times less bitrate).

Finally, since there are many modern codecs available nowadays we also test our method for different codecs, which not always share artifact behavior with JPEG. In particular we considered WebP, JPEG2000 and BPG. We tuned all codecs to obtain the same average bitrate on the whole VOC2007 dataset of the respective JPEG codec using a QF of 20. Results are reported in Table III, and show that our novel approach is effective also for all these compression algorithms.

Additional comparison in terms of mAP for the task of object detection is reported in Table V, where a subset of PASCAL VOC 2007 has been used, following the experimental setup of [51]. Our proposed GAN method obtains a better result than the current state-of-the-art.

2) *Experiments on MS-COCO*: In Figure 7 we show how mean Average Precision (mAP) varies on the MS-COCO test set. When aggressive compression is used GAN_{L_1} and GAN_{VGG} get the best results, while the simpler AR-CNN is





















											
JPEG 20	58.7	69.2	51.6	43.4	35.0	67.3	71.0	55.9	33.4	55.9	57.9
AR-CNN [9]	64.1	68.6	52.3	41.3	36.7	70.2	74.2	53.0	36.3	57.4	60.7
MSE	64.7	69.6	51.2	40.6	40.9	71.3	75.0	54.2	38.6	54.6	61.4
Our SSIM	65.5	70.6	51.3	41.7	41.1	71.3	74.6	55.5	38.7	53.8	61.5
Our GAN-Y	65.7	69.6	54.7	46.1	35.4	71.9	70.8	67.3	38.0	65.3	60.5
Our GAN-L1	64.4	75.0	52.4	42.1	42.7	69.1	75.5	66.7	40.2	61.6	59.7
Our GAN-VGG	66.6	75.3	56.5	47.5	39.5	72.7	77.0	72.5	40.3	68.4	60.2
Our GAN-VGG-BN	65.4	78.7	57.4	50.2	39.9	72.7	77.1	76.4	42.8	70.0	60.2
Original	69.8	78.8	69.2	55.9	48.8	76.9	79.8	85.8	48.7	76.2	63.7
										mAP	
JPEG 20	53.2	69.1	66.5	63.8	26.0	48.2	43.4	70.7	57.0	54.9	
AR-CNN [9]	58.1	72.4	66.1	65.8	31.3	49.9	52.6	71.2	57.8	57.0	
Our MSE	59.5	71.3	66.8	66.4	31.0	48.5	52.2	67.6	60.0	57.3	
Our SSIM	59.6	72.0	66.6	66.3	30.8	48.2	53.2	66.8	59.8	57.4	
Our GAN-Y	68.1	73.8	66.1	66.2	29.0	60.8	54.4	72.2	60.0	59.8	
Our GAN-L1	67.9	74.9	66.6	66.4	30.9	54.3	58.7	65.5	61.3	59.8	
Our GAN-VGG	71.8	75.3	70.7	67.0	30.3	62.5	58.6	71.2	61.1	62.3	
Our GAN-VGG-BN	72.4	77.4	72.3	68.0	32.0	56.8	58.4	71.7	63.3	63.1	
Original	79.0	80.2	75.7	76.3	37.6	68.3	67.2	77.7	66.7	69.1	

TABLE IV: Object detection performance measured as mean average precision (mAP) on PASCAL VOC2007 for different reconstruction algorithms. Bold numbers indicate best results.

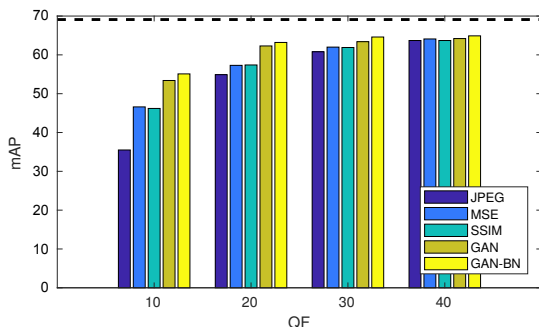


Fig. 6: Mean average precision (mAP), for different Quality Factors (QF), and restoration approaches, on PASCAL VOC2007.

Method	mAP
JPEG 10	35.9
AR-CNN-Y[9]	42.9
SA-DCT[12]	48.5
Our MSE[13]	51.9
ED [51]	52.5
CED-EST [51]	52.6
CED-GT [51]	55.0
Our GAN	55.6
Original	70.5

TABLE V: Object detection performance measured on the subset of PASCAL VOC 2007 dataset used in [51].

less effective. For higher QF values we do not observe such difference, if AP is measured on all 80 classes. Interestingly, looking at classes separately we can see that for certain classes compression artifacts degrade more AP. This is shown in Tab. VI, where we report the 5 classes that obtain the highest and the lowest improvements in performance using GAN_{VGG}.

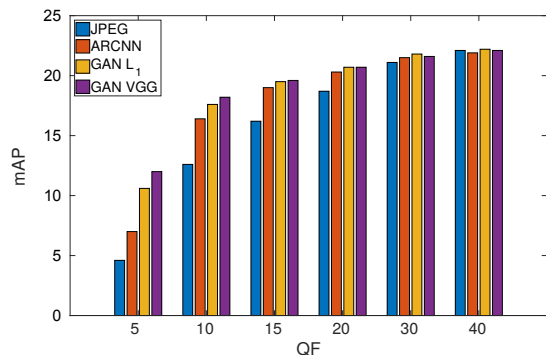


Fig. 7: Mean Average Precision on MS-COCO varying the QF (the higher, the better). For high compression rates GAN methods get the best results. For QFs higher than 30, the variation is minimal.

It can be noticed that among the 5 classes that obtain the largest improvements there are several animals (e.g. *cat*, *dog*, *bear*, etc.): this is due to the reconstruction of finer details like fur obtained using the proposed GAN approach.

3) Evaluation of compression effects for object detection:

To gain insight on the behavior of semantic computer vision algorithms on compressed and reconstructed images, we analyze how deep convolutional features vary under image compression, and how this variation is moderated when artifact removal techniques are applied. We run the following test on MS-COCO, for every quality factor and method involved in our study, we compute the mean relative error of each layer of the Faster R-CNN detector [38]:

$$\epsilon_l = \frac{|\phi_l(I^R) - \phi_l(I)|}{\phi_l(I)} \quad (11)$$

	QF=5		QF=10		QF=15		QF=20		QF=30		QF=40	
Highest 5 gains	pizza	24.9	cat	25.9	cat	20.3	cat	13.5	cat	5.3	tv	2.5
	bear	21.5	bear	25.3	couch	12.8	couch	9.2	couch	3.6	cat	2.4
	firehydrant	20.8	elephant	21.0	dog	11.6	bear	7.2	mouse	3.1	couch	1.5
	giraffe	20.1	dog	17.1	bear	11.3	dog	6.7	toilet	2.8	mouse	1.4
	elephant	20.0	toilet	14.7	toilet	9.7	toilet	5.9	microwave	2.6	laptop	1.3
Lowest 5 gains	hairdrier	0.0	train	0.0	train	-0.6	giraffe	-0.5	train	-1.6	train	-2.1
	handbag	0.2	hairdrier	0.1	bus	-0.1	keyboard	-0.4	bus	-1.4	firehydrant	-2.0
	toaster	0.4	toaster	0.4	hairdrier	0.0	baseballbat	-0.1	giraffe	-1.4	broccoli	-1.2
	book	0.4	book	0.7	scissors	0.1	train	-0.1	baseballbat	-1.3	elephant	-1.0
	spoon	0.9	handbag	0.8	carrot	0.2	bicycle	0.1	broccoli	-1.2	bear	-1.0

TABLE VI: Most and least affected classes in terms of AP for different QF values when using GAN_{VGG} method to eliminate compression artifacts.

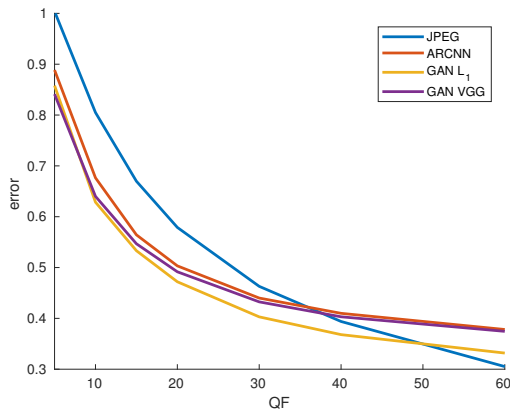


Fig. 8: Mean relative error, averaged over all layers, for different QF and artifact removal techniques (the lower, the better). The proposed GAN restoration approach with L_1 loss obtains the smallest error; using VGG loss still improves over AR-CNN.

where $\phi_l(\cdot)$ are feature maps for layer l .

Results are reported in the plots of Fig. 8, that shows the mean relative error, averaged over all layers, for different QF values. For higher QF values JPEG compression affects little, but noticeably feature map values. The variation is closer to 30% for QF=60, and applying reconstruction methods on high quality images, as expected, does not produce any benefit. Clearly, when QF become smaller all reconstruction techniques help in generating images with feature maps closer to the original one, with GAN $_{L_1}$ obtaining the best results and becoming effective from QF=50. The novel GAN approach obtains better results than AR-CNN also using VGG loss, but it is particularly effective when using L_1 loss for QF ≥ 20 .

In Fig. 9 we analyze the behavior for all feature maps, reporting the mean relative error for all the layers and different QF values. It is interesting to note that the first and last layers are less affected, while the ones that exhibit the most relative error are `conv3_2` and `conv4_2`. As also shown in Fig. 8, applying reconstruction is not beneficial for QF=60, while for other QF values it can be seen that the error is reduced for all layers, and specifically for the ones which are most affected. Notably, highest average relative errors can reach 100% \sim 150%.

We can conclude that applying image restoration to images

improves the fidelity of CNN feature maps. Nonetheless the behavior of our GAN models appear close to that of AR-CNN which has instead much worse performance in terms of mAP. Therefore we perform further experiments to understand the relation of feature map error and object detection quality. In particular, we measure, for each class, how much the drop in average precision depends from image corruption. In Fig. 10 we show, for all the analyzed QF values, a scatter plot of ΔAP_c and $\bar{\epsilon}^c$ for each class c . Where

$$\Delta AP_c = \frac{AP_c - AP_c^R}{AP_c} \quad (12)$$

is the relative drop in average precision when detection is performed on original images (I) and restored images (I^R), with a special case of JPEG, when image reconstruction is not performed at all and

$$\bar{\epsilon}^c = \frac{1}{|L|} \sum_{l \in L} \epsilon_l^c \quad (13)$$

is the error averaged over the set of layers L for a class c . The lower the ΔAP_c , the better the performance of the classifier and of the reconstruction algorithm.

As shown in Fig. 10, there is an interesting correlation between feature map error and AP drop per class. Indeed, the error presented by feature maps, negatively affects performance in terms of average precision, in case no reconstruction is applied. Interestingly when using our GAN based method it can be seen that feature map error is still present, but with little correlation with ΔAP , even for extremely aggressive compression rates (e.g. QF=5, 10). This means that the reconstruction process will yield images that are different from their original uncompressed version and this is reflected in the error of feature maps. Nonetheless, image appearance in terms of semantic content is improved, therefore leading to a lower drop in AP.

D. QF Predictor and Multi-QF evaluation

We want to understand how our QF compression predictor helps to improve the GAN reconstruction when the quality factor of the compressed image is not known. In the first place, we evaluate the performance of QF selector, judging its classification capabilities. For this experiment, we have

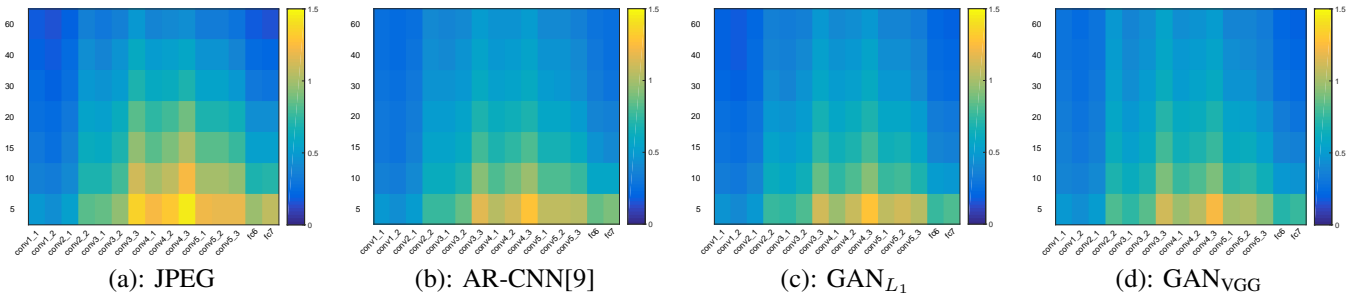


Fig. 9: Mean relative error, for all layers, for different QF and artifact removal techniques (the lower, the better); the proposed GAN approach with L_1 loss obtains the least error.

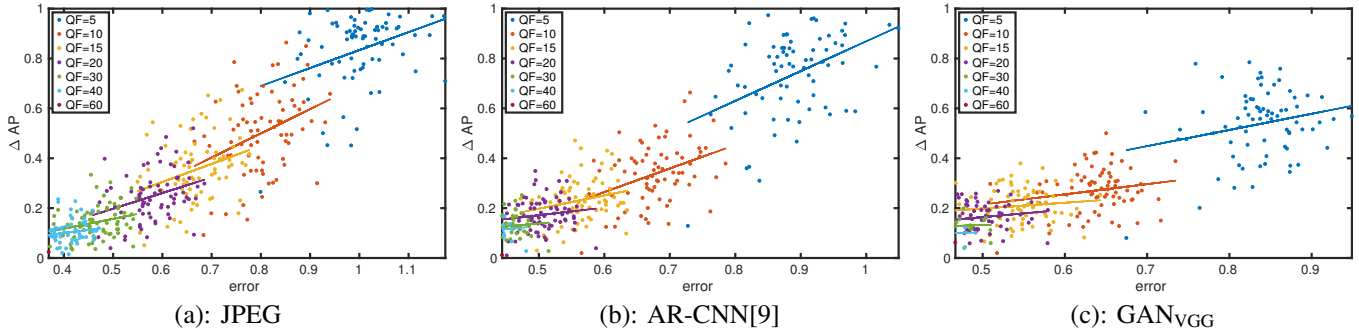


Fig. 10: Correlation of drop in AP with average feature map error for different QF and methods. GAN based methods attain lowest error and AP drop.

compressed the whole PASCAL VOC 2007 dataset at six different QFs (5,10,20,30,40,60). We evaluate the QF predictor as a classifier, rounding the regressor output. We report the classification performance as a normalized confusion matrix in Fig. 11. Interestingly the accuracy is extremely high and misclassification are exclusively on close-by factors. In Tab. VII we analyze the performance of our networks trained on patches of multiple QF (Multi-QF) and the results obtained by single QF networks driven by our predictor (QF Predictor). As reference performance bounds, we report also the results on the compressed image (JPEG) and the results obtained by always using the right QF GAN (Oracle). Evaluation is carried on in terms of PSNR, PSNR-B and SSIM on LIVE1 and BSD500 datasets, at different QF values. It can be observed that using our proposed QF predictor always improves over both the JPEG baseline and the Multi-QF approach, resulting in figures that are on par with a QF oracle, except for QF=40.

Then, we investigate the behavior of single QF models inside the ensemble by applying them to images of different QFs. In Fig. 12 we measure mAP for images compressed at various QFs and reconstructed with QF specific models. It can be seen that when we use models for similar QFs mAP varies smoothly. Interestingly, for images with lower QFs such as 5 and 10 we see improvements for every model applied. Higher quality images must be restored with models trained with higher QFs, otherwise performance can even degrade.

Finally, we evaluate the restoration of images using our full system for the task of object detection on PASCAL VOC 2007. In Tab. VIII we compare the results obtained with Multi-QF with the results obtained by our QF Predictor. Again, we report

Predicted QF	5	97.5	2.2	0.3	0	0	0
	10	7	92.6	0.4	0	0	0
	20	0	0.3	99.4	0.3	0	0
	30	0	0	1.8	97.7	0.5	0
	40	0	0	0	5.4	94.5	0.2
	60	0	0	0	0	1.4	98.6
		5	10	20	30	40	60
		Target QF					

Fig. 11: QF compression predictor - confusion matrix.

also the results on the compressed image (JPEG, as lower bound) and the results obtained by a QF Oracle (upper bound).

Our ensemble is always outperforming the Multi-QF model attaining performance very close to the oracle. When looking at some qualitative results, shown in Fig. 13, it can be seen that the Multi-QF approach is able to recover from most artifacts but it is also responsible for the introduction of checkerboard artifacts as in the third row. In general the Multi-QF model generates softer looking images with fewer details, as can be observed in the first and fourth lines of Fig. 13.

Using oracle driven ensembles should always yield the best result. From Tab. VIII it can be seen that the mAP figures for

QF	Method	LIVE1			BSD500		
		PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
10	JPEG	27.77	25.33	0.791	27.58	24.97	0.769
	Multi-QF MSE	29.45	29.10	0.834	29.03	28.61	0.807
	Multi-QF SSIM	28.94	28.46	0.840	28.52	27.93	0.816
	Multi-QF GAN	27.29	26.69	0.773	27.01	26.30	0.746
	QF Predictor MSE	29.47	29.13	0.833	29.05	28.64	0.806
	QF Predictor SSIM	28.94	28.46	0.841	28.53	27.97	0.816
	QF Predictor GAN	27.65	27.63	0.777	27.31	27.28	0.749
	Oracle MSE	29.47	29.13	0.833	29.05	28.64	0.806
	Oracle SSIM	28.94	28.46	0.841	28.53	27.97	0.816
	Oracle GAN	27.65	27.63	0.777	27.31	27.28	0.749
20	JPEG	30.07	27.57	0.868	29.72	26.97	0.852
	Multi-QF MSE	31.77	31.26	0.896	31.20	30.48	0.876
	Multi-QF SSIM	31.38	30.77	0.900	30.79	29.92	0.882
	Multi-QF GAN	28.35	28.1	0.817	28.07	27.76	0.794
	QF Predictor MSE	31.81	31.29	0.897	31.23	30.49	0.877
	QF Predictor SSIM	31.51	30.84	0.901	30.92	30.01	0.883
	QF Predictor GAN	29.99	29.69	0.864	29.48	29.03	0.841
	Oracle MSE	31.81	31.29	0.897	31.23	30.49	0.877
	Oracle SSIM	31.51	30.84	0.901	30.92	30.01	0.883
	Oracle GAN	29.99	29.69	0.864	29.48	29.03	0.841
30	JPEG	31.41	28.92	0.900	30.98	28.23	0.886
	Multi-QF MSE	33.15	32.51	0.922	32.44	31.41	0.906
	Multi-QF SSIM	32.87	32.09	0.925	32.15	30.97	0.909
	Multi-QF GAN	28.58	28.75	0.832	28.50	28.00	0.811
	QF Predictor MSE	33.21	32.51	0.923	32.54	31.58	0.907
	QF Predictor SSIM	32.95	32.17	0.926	32.25	31.15	0.910
	QF Predictor GAN	31.65	31.38	0.900	31.02	30.55	0.881
	Oracle MSE	33.21	32.51	0.923	32.53	31.57	0.906
	Oracle SSIM	32.95	32.17	0.926	32.26	31.16	0.911
	Oracle GAN	31.65	31.38	0.900	31.04	30.57	0.881
40	JPEG	32.35	29.96	0.917	31.88	29.14	0.906
	Multi-QF MSE	34.09	33.40	0.935	33.30	32.18	0.921
	Multi-QF SSIM	33.82	33.00	0.937	33.04	31.72	0.924
	Multi-QF GAN	28.99	28.84	0.837	28.61	28.20	0.815
	QF Predictor MSE	34.13	33.37	0.936	33.38	32.28	0.922
	QF Predictor SSIM	33.95	33.04	0.938	33.17	31.88	0.925
	QF Predictor GAN	31.64	31.18	0.903	30.99	30.20	0.883
	Oracle MSE	34.17	33.42	0.937	33.45	32.34	0.923
	Oracle SSIM	33.98	33.07	0.939	33.25	31.94	0.926
	Oracle GAN	31.64	31.17	0.903	30.98	30.16	0.884

TABLE VII: Average PSNR, PSNR-B and SSIM results on BDS500 and LIVE1 using different reconstruction approaches. Evaluation using luminance. The proposed approach based on QF Predictor is typically on par with the use of an oracle.

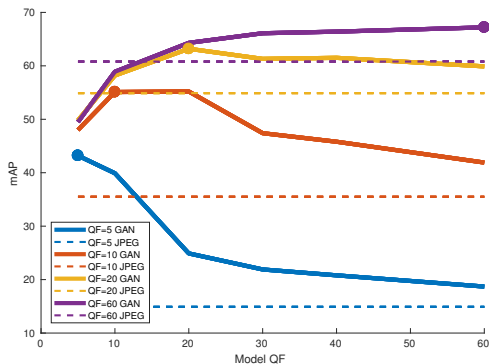


Fig. 12: Mean Average Precision on PASCAL VOC2007 varying image QF as well as single QF GANs. Original mAP reported as dashed lines for every QF. Circular markers indicate the maximum mAP, obtained with the correct model.

Oracle and QF Predictor are very close ($\pm 0.1\%$) except for QF=10 where Oracle has 0.8% more. Looking at Fig. 11 it can be seen that most of the errors occur for images at QF=10, in particular 7% of the images are classified as QF=5. In Fig. 12 it can be seen that when using a model trained for QF=5 on images obtained with a higher QF there is always a decrease in mAP.

The complementary behavior can be observed for QF=5 in Tab. VIII where the Oracle obtains 0.3% less than the QF Predictor. According to Fig. 11, 2.5% of the samples

compressed with QF=5 are misclassified as 10 and 20. We measure mAP on this smaller set comparing the GAN trained for QF=5 and GANs selected by the QF Predictor. The first obtains 54.9% while the latter 64.1%, showing that this kind of “misclassification” may even be beneficial. A similar behavior, although less pronounced, happens for QF=20-60.

	5	10	20	30	40	60
JPEG	14.9	35.5	54.9	60.8	63.7	66.3
Multi-QF	37.2	54.0	62.7	64.6	65.0	65.7
QF Predictor	43.5	54.3	63.1	64.7	65.0	67.3
Oracle	43.2	55.1	63.2	64.6	64.9	67.2

TABLE VIII: Mean average precision on PASCAL VOC2007 with different Multi QF approaches.

E. Segmentation Mask Proposal

In this experiment we analyze the performance of the generation of mask proposals for an image on MS-COCO[29] using 20,000 images on the *test-dev* set as in Sect.V-C. These proposals should precisely segment objects in a scene. Mask proposals can be used to derive bounding boxes to be fed to an object detector. Mask proposals, once evaluated by a classifier, can be used to label image pixels with categories. Differently from semantic segmentation, modern benchmarks evaluate not just the label correctness pixel-wise but also instance-wise, meaning that multiple people close-by should not be assigned a single “person” mask.

1) *Method*: Also in this experiment we use a recent method based on deep neural networks, i.e. SharpMask [37]. This approach is based on a previous method, proposed by the same authors named DeepMask [36], which learns to generate a binary mask jointly optimizing two logistic regression losses: a patch-wise object presence loss and a pixel-wise mask loss. Mask loss is inactive when an object is not present inside the patch. SharpMask proposes a refinement process able to improve 10-20% in object mask accuracy. Both methods use a pre-trained VGG-16 network to extract features.

We test SharpMask [37], with the same protocol described in Sect. V-C. We measure performance in terms of Average Recall for 10 proposals. This means that we average object recall over a set of intersection over union values, and report looking only at the first 10 proposals of every image (AR@10). Similarly to results reported in Sect. V-C we have GAN_{VGG} obtaining the best performance in recovering from artifacts. This behavior is consistent for all QFs. Images compressed with a QF higher than 40 exhibit little loss in AR@10.

F. Subjective evaluation

In this experiment we assess how images obtained with the proposed methods are perceived by a human viewer, evaluating in particular the preservation of details and overall quality of an image using the SSIM loss and the GAN-based approaches. 10 viewers have participated to the test, a number that is considered enough for subjective image quality evaluation tests [47]; no viewer was familiar with image quality evaluation or the approaches proposed in this work. A Double-Stimulus Impairment Scale (DSIS) experimental

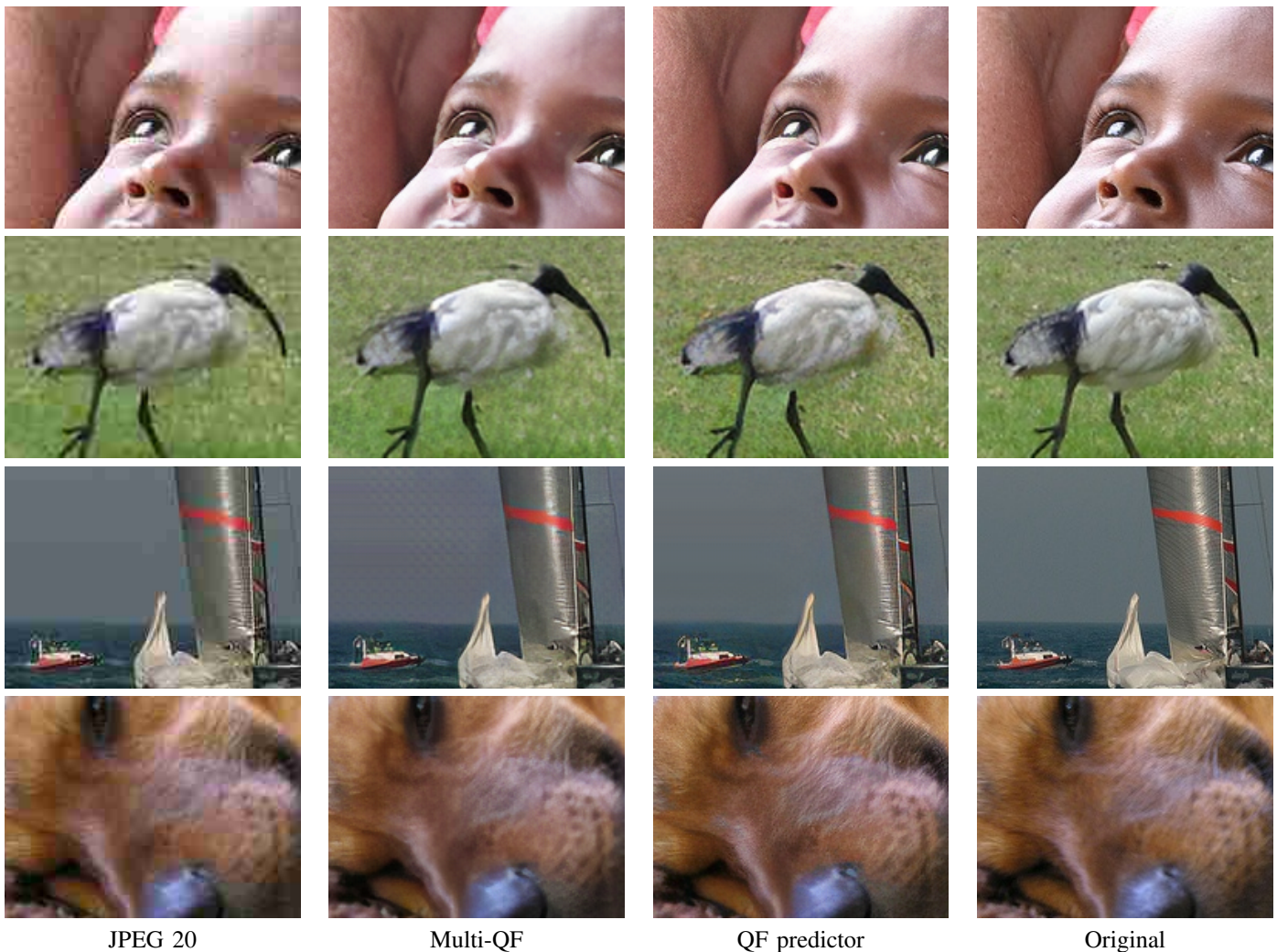


Fig. 13: Qualitative comparison in the multiple QF setting.

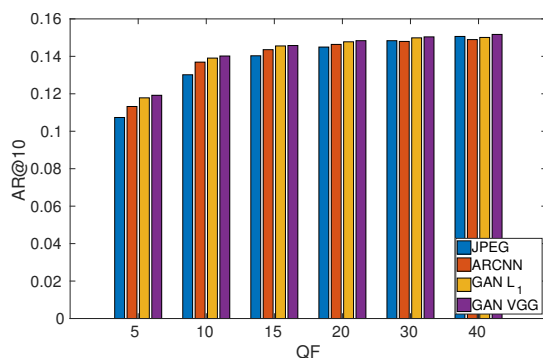


Fig. 14: Average Recall for 10 proposals per image for different QF and methods. Performance at low QFs for GAN based methods is superior.

setup has been developed using VQone, a tool designed to perform subjective image quality evaluations [34]. We asked participants to evaluate reconstructed images comparing them to the original uncompressed ones with a grade of similarity on a continuous scale from 0 to 100. A slider with no marked

values is used in order to avoid preferred numbers. A set of 50 images is randomly selected from the BSD500 dataset. Considering our estimation of test completion time we chose this amount of images to keep each session under 30 minutes as recommended by ITU-R BT.500-13 [19].

The selected images contain different subjects, such as persons, animals, man-made objects, nature scenes, etc. For each original image have been shown both an image processed with the SSIM loss network and the GAN network, resulting in an overall collection of 1,000 judgements. The order of appearance of the images was randomized to avoid showing the results of the two approaches always in the same order; we also randomized the order of presentation of the tests for each viewer. In Table IX are reported final results as MOS (Mean Opinion Scores) with standard deviation. They show that the GAN-based network is able to produce images that are perceptually more similar to the original ones. In Fig 15 we report MOS for each image with a 95% confidence interval. It appears clearly that in roughly 90% of the cases our GAN-based network restored images are considered more similar to the original with respect to the one using the SSIM-based loss.

Method	MOS	std. dev.
Our SSIM	49.51	22.72
Our GAN	68.32	20.75

TABLE IX: Subjective image quality evaluation in terms of Mean Opinion Score (MOS) on 50 images of BSD500 dataset.

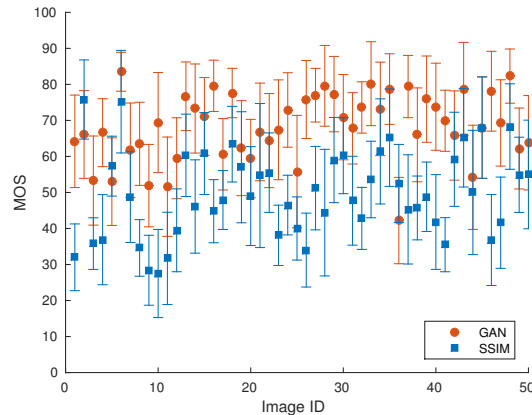


Fig. 15: MOS values, with 0.95 confidence, for all the 50 images used in the subjective evaluation.

VI. CONCLUSION

We have shown that compression artifact removal can be performed by learning an image transformation task with a deep residual convolutional neural network. We show that conditional Generative Adversarial Networks produce higher quality images with sharp details which are relevant not only to the human eye but also for semantic computer vision tasks. Our model, trained by minimizing SSIM based loss obtains state of the art results according to standard image similarity metrics. Nonetheless, images reconstructed as such appear blurry and missing details at higher frequencies. Our GAN, trained alternating full size patch generation with sub-patch discrimination solve this issue.

Considering that compression parameters are not known in advance, we propose a method which is able to predict the quality of the image with high accuracy and pick a specialized GAN model out of an ensemble to restore the image, obtaining results on par with the same ensemble driven by an oracle.

We have extensively analyzed the behavior of deep CNN based algorithms when processing images that are compressed, evaluating results at different compression levels. As expected artifacts appearing even at moderately compression rates modify feature maps. This phenomenon is shown to correlate with errors in semantic tasks such as object detection and segmentation. We have shown a high drop in performance for classes where texture is an important cue and entities are deformable and articulated, such as cats and other animals.

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.

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

REFERENCES

- [1] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proc. of CVPR Workshops*, 2017.
- [2] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *Proc. of ICLR*, 2018.
- [3] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. *CoRR*, abs/1511.05666, 2015.
- [4] L. Cavigelli, P. Hager, and L. Benini. CAS-CNN: A deep convolutional neural network for image compression artifact suppression. In *Proc. of IJCNN*, 2017.
- [5] H. Chang, M. K. Ng, and T. Zeng. Reducing artifacts in JPEG decomposition via a learned dictionary. *IEEE Transactions on Signal Processing*, 62(3):718–728, Feb 2014.
- [6] H. Chen, X. He, L. Qing, and Q. Teng. Single image super-resolution via adaptive transform-based nonlocal self-similarity modeling and learning-based gradient regularization. *IEEE Transactions on Multimedia*, 19(8):1702–1717, 2017.
- [7] R. Dahl, M. Norouzi, and J. Shlens. Pixel Recursive Super Resolution. *ArXiv preprint arXiv:1702.00783*, Feb. 2017.
- [8] 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.
- [9] C. Dong, Y. Deng, C. Change Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *Proc. of ICCV*, 2015.
- [10] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. of NIPS*, 2016.
- [11] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.
- [12] 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, 2007.
- [13] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo. Deep generative adversarial compression artifact removal. In *Proc. of ICCV*, 2017.
- [14] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *CoRR*, abs/1505.07376, 2015.
- [15] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. of CVPR*, 2016.
- [16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. of NIPS*, 2014.
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of CVPR*, 2016.
- [18] X. He, Q. Hu, X. Zhang, C. Zhang, W. Lin, and X. Han. Enhancing HEVC compressed videos with a partition-masked convolutional neural network. In *Proc. of ICIP*, 2018.
- [19] ITU. *Rec. ITU-R BT.500-13 - Methodology for the subjective assessment of the quality of television pictures*, 2012.
- [20] V. Jakhetiya, W. Lin, S. P. Jaiswal, S. C. Guntuku, and O. C. Au. Maximum a posteriori and perceptually motivated reconstruction algorithm: A generic framework. *IEEE Transactions on Multimedia*, 19(1):93–106, 2017.
- [21] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. of ECCV*, 2016.
- [22] 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.
- [23] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proc. of CVPR*, 2016.
- [24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.
- [25] N. Kumar and A. Sethi. Super resolution by comprehensively exploiting dependencies of wavelet coefficients. *IEEE Transactions on Multimedia*, 20(2):298–309, 2018.
- [26] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.
- [27] T. Li, X. He, L. Qing, Q. Teng, and H. Chen. An iterative framework of cascaded deblocking and super-resolution for compressed images. *IEEE Transactions on Multimedia*, 2017.
- [28] Y. Li, F. Guo, R. T. Tan, and M. S. Brown. A contrast enhancement framework with JPEG artifacts suppression. In *Proc. of ECCV*, 2014.
- [29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proc. of ECCV*, 2014.

- [30] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of CVPR*, 2015.
- [31] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Proc. of NIPS*, 2016.
- [32] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. of ICCV*, 2001.
- [33] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [34] M. Nuutinen, T. Virtanen, O. Rummukainen, and J. Häkkinen. VQone MATLAB toolbox: A graphical experiment builder for image and video quality evaluations. *Behavior Research Methods*, 48(1):138–150, 2016.
- [35] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. <http://distill.pub/2016/deconv-checkerboard>.
- [36] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *Proc. of NIPS*, 2015.
- [37] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *Proc. of ECCV*, 2016.
- [38] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. of NIPS*, 2015.
- [39] O. Rippel and L. Bourdev. Real-time adaptive image compression. In *Proc. of ICML*, 2017.
- [40] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. *CoRR*, abs/1606.03498, 2016.
- [41] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. LIVE Image Quality Assessment Database Release 2, Apr. 2014.
- [42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR*, 2015.
- [43] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik. Compression artifacts removal using convolutional neural networks. *arXiv preprint arXiv:1605.00366*, 2016.
- [44] Z. Wang, A. C. Bovik, and L. Lu. Why is image quality assessment so difficult? In *Proc. of ICASSP*, 2002.
- [45] 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, April 2004.
- [46] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang. D3: Deep dual-domain based fast restoration of JPEG-compressed images. In *Proc. of CVPR*, 2016.
- [47] S. Winkler. On the properties of subjective ratings in video quality experiments. In *Proc. of QME*, 2009.
- [48] 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, 2009.
- [49] S. Yang, S. Kittitornkun, Y.-H. Hu, T. Q. Nguyen, and D. L. Tull. Blocking artifact free inverse discrete cosine transform. In *Proc. of ICIP*, 2000.
- [50] C. Yim and A. C. Bovik. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, 2011.
- [51] J. Yoo, S.-h. Lee, and N. Kwak. Image restoration by estimating frequency distribution of local patches. In *Proc. of CVPR*, 2018.
- [52] J. Zhang, R. Xiong, C. Zhao, Y. Zhang, S. Ma, and W. Gao. CON-COLOR: Constrained non-convex low-rank model for image deblocking. *IEEE Transactions on Image Processing*, 25(3):1246–1259, March 2016.
- [53] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26:3142–3155, 2017.
- [54] 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.



Leonardo Galteri received a master degree magna cum laude in computer engineering from the University of Florence in 2014 and PhD in 2018 with a thesis on semantic video compression and object detection. Currently he is a postdoc at the Media Integration and Communication Center of the University of Florence. His research interest focus on object detection, visual saliency and video compression.



Lorenzo Seidenari is currently an Assistant Professor at the Media Integration and Communication Center of the University of Florence. He received his Ph.D. degree in computer engineering in 2012 from the University of Florence. His research focuses on deep learning for object and action recognition in video and images. On this topics he addressed RGB-D activity recognition, embedding learning for multimodal-fusion, anomaly detection in video and people behavior profiling. He was a visiting scholar at the University of Michigan in 2013. He organized and gave a tutorial at ICPR 2012 on image categorization. He is author of 12 journal papers and more than 30 peer-reviewed conference papers. He has an h-index of 17 with more than 900 citations. He is Associate Editor of Multimedia Tools and Applications.



Marco Bertini received the Laurea degree in Electronic Engineering from the University of Florence in 1999, and Ph.D. in 2004. He is working at the Media Integration and Communication Center of the University of Florence and is Associate Professor at the School of Engineering of the University of Florence. His interests are focused on digital libraries, multimedia databases and social media. On these subjects he has addressed semantic analysis, content indexing and annotation, semantic retrieval and transcoding. He is author of 22 journal papers and more than 100 peer-reviewed conference papers, with h-index: 24 (according to Google Scholar). He is associate editor of IEEE Transactions on Multimedia.



Alberto Del Bimbo is a Full Professor of Computer Engineering, and the Director of the Media Integration and Communication Center at the University of Florence, Italy. His scientific interests are computer vision and multimedia. He was the President of the IAPR Italian Chapter and Member-at-Large of the IEEE Publication Board. He acted as General Chair of ACM Multimedia 2010 and ECCV2012 the European Conference on Computer Vision. He is IAPR Fellow and ACM Distinguished Scientist. In 2016 he received the SIGMM Technical Achievement Award for Outstanding Technical Contributions to Multimedia Computing, Communications and Applications. He was Associate Editor of Pattern Recognition, IEEE Transactions on Multimedia, and IEEE Transactions on Pattern Analysis and Machine Intelligence. Presently he is Editor-in-Chief of ACM Transactions on Multimedia Computing, Communications, and Applications and Associate Editor of Multimedia Tools and Applications and Pattern Analysis and Applications.