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.

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.
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 deblock- ing 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 as to ease the training. Yoo et al. [51] aim at restoring high-frequency details in JPEG compressed images employing and 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 $8 \times 8$ 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 lossy compression; it also permits a first step of separating luminance from chrominance information. 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 $3\times3$ 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 larger batches. We are interested in removing artifacts at scales close to the block size. For this reason we can learn models on smaller patches using larger batches.

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

$$I^C = A(I, QF) \in [0, 255]^{W \times H \times C}$$  \hspace{1cm} (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 $G \approx A^{-1}_{QF}$ that reconstructs $I$ from $I^C$:

$$G(I^C) = I^R \approx I$$  \hspace{1cm} (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:

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

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

$$\theta_g = \{W_{1:K}; b_{1:K}\}$$  \hspace{1cm} (4)

$$l_{AR}(I, G(I^C, \theta_g))$$  \hspace{1cm} (5)

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 $3\times3$ 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 larger batches. We are interested in removing artifacts 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.

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

$$\theta_g = \{W_{1:K}; b_{1:K}\}$$  \hspace{1cm} (4)

$$l_{AR}(I, G(I^C, \theta_g))$$  \hspace{1cm} (5)

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 $3\times3$ 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 larger batches. We are interested in removing artifacts 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 $3\times3$ 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 larger batches. We are interested in removing artifacts 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 $3\times3$ 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 larger batches. We are interested in removing artifacts at scales close to the block size. For this reason we can learn models on smaller patches using larger batches.
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):

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

This loss is commonly used in image reconstruction and restoration tasks [9], [43], [31]. It has been shown that $l_{\text{MSE}}$ 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:

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

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

$$l_{\text{SSIM}} = -\frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \text{SSIM} (I_{x,y}, I_{x,y}^R)$$ (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 are able to “fool” the discriminator. The distance between images is computed after projecting $I$ and $I^R$ on a feature space by some differentiable function $\phi$ and taking the Euclidean distance between the two feature representations:

$$l_p = \frac{1}{W_fH_f} \sum_{x=1}^{W_f} \sum_{y=1}^{H_f} (\phi (I)_{x,y} - \phi (I^R)_{x,y})^2$$ (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:

$$l_{\text{AR}} = l_p + \lambda l_{\text{adv}}.$$ (9)

Where $l_{\text{adv}}$ is the standard adversarial loss:

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

that rewards solutions that are able to “fool” the discriminator.
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 $128 \times 128$ patches.

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 $128 \times 128$ 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 methods, we adopted the 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. 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
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 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 outperform 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× 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\(_{VGG}^2\) and GAN\(_{VGG}\) get the best results, while the simpler AR-CNN is
GALTERI et al.: DEEP UNIVERSAL GENERATIVE ADVERSARIAL COMPRESSION ARTIFACT REMOVAL

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG 20</td>
<td>58.7</td>
</tr>
<tr>
<td>AR-CNN [9]</td>
<td>64.1</td>
</tr>
<tr>
<td>MSE</td>
<td>64.7</td>
</tr>
<tr>
<td>Our SSIM</td>
<td>65.5</td>
</tr>
<tr>
<td>Our GAN-Y</td>
<td>65.7</td>
</tr>
<tr>
<td>Our GAN-L1</td>
<td>64.4</td>
</tr>
<tr>
<td>Our GAN-VGG</td>
<td>66.6</td>
</tr>
<tr>
<td>Our GAN-VGG-BN</td>
<td>65.4</td>
</tr>
<tr>
<td>Original</td>
<td>69.8</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG 20</td>
<td>53.2</td>
</tr>
<tr>
<td>AR-CNN [9]</td>
<td>58.1</td>
</tr>
<tr>
<td>Our MSE</td>
<td>59.5</td>
</tr>
<tr>
<td>Our SSIM</td>
<td>59.6</td>
</tr>
<tr>
<td>Our GAN-Y</td>
<td>68.1</td>
</tr>
<tr>
<td>Our GAN-L1</td>
<td>67.9</td>
</tr>
<tr>
<td>Our GAN-VGG</td>
<td>71.8</td>
</tr>
<tr>
<td>Our GAN-VGG-BN</td>
<td>72.4</td>
</tr>
<tr>
<td>Original</td>
<td>79.0</td>
</tr>
</tbody>
</table>

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

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]:

$$\epsilon_I = \left| \frac{\phi_I(I^R) - \phi_I(I)}{\phi_I(I)} \right|$$  (11)
Table VI: Most and least affected classes in terms of AP for different QF values when using $\text{GAN}_{\text{VGG}}$ method to eliminate compression artifacts.

<table>
<thead>
<tr>
<th>Highest 5 gains</th>
<th>Lowest 5 gains</th>
</tr>
</thead>
<tbody>
<tr>
<td>QF=5</td>
<td>QF=5</td>
</tr>
<tr>
<td>pizza</td>
<td>hairdrier</td>
</tr>
<tr>
<td>bear</td>
<td>train</td>
</tr>
<tr>
<td>cat</td>
<td>book</td>
</tr>
<tr>
<td>24.9</td>
<td>hairdrier</td>
</tr>
<tr>
<td>25.9</td>
<td>0.0</td>
</tr>
<tr>
<td>20.3</td>
<td>0.2</td>
</tr>
<tr>
<td>cat</td>
<td>toaster</td>
</tr>
<tr>
<td>13.5</td>
<td>0.4</td>
</tr>
<tr>
<td>cat</td>
<td>book</td>
</tr>
<tr>
<td>5.3</td>
<td>0.4</td>
</tr>
<tr>
<td>tv</td>
<td>spoon</td>
</tr>
<tr>
<td>2.5</td>
<td>0.9</td>
</tr>
</tbody>
</table>

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 $\text{GAN}_{L_1}$ 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 $\text{conv}_3$ and $\text{conv}_4$. 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% ~ 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 $\Delta \text{AP}_c$ and $\bar{e}_c$ for each class $c$. Where

$$\Delta \text{AP}_c = \frac{\text{AP}_c - \text{AP}^R_c}{\text{AP}_c}$$

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

$$\bar{e}_c = \frac{1}{|L|} \sum_{l \in L} e_{c_l}$$

is the error averaged over the set of layers $L$ for a class $c$. The lower the $\Delta \text{AP}_c$, 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 $\Delta \text{AP}$, 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
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 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...
Oracle and QF Predictor are very close (±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.

![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.](image)

Oracle and QF Predictor are very close (±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.

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).

![Oracle MSE](image)

Similarly to results reported in Sect. V-C we have GANVGG 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. 14: Average Recall for 10 proposals per image for different QF and methods. Performance at low QFs for GAN based methods is superior.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>MOS</th>
<th>std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our SSIM</td>
<td>49.51</td>
<td>22.72</td>
</tr>
<tr>
<td>Our GAN</td>
<td>68.32</td>
<td>20.75</td>
</tr>
</tbody>
</table>

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


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-based retrieval 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 (ac- cording 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 ECCV 2012 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.