IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 3, 2020, accepted March 23, 2020, date of publication March 30, 2020, date of current version April 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984387

Deep Residual Autoencoder for Blind Universal JPEG Restoration

SIMONE ZINI, SIMONE BIANCO, AND RAIMONDO SCHETTINI
Department of Informatics, Systems, and Communication, University of Milano–Bicocca, 20126 Milan, Italy
Corresponding author: Simone Zini (s.zini1@campus.unimib.it)

ABSTRACT We propose a deep residual autoencoder exploiting Residual-in-Residual Dense Blocks (RRDB) leveraging both the learning capacity of deep residual networks and prior knowledge of the JPEG compression pipeline. The proposed method is blind and universal, consisting of a unique model that effectively restores images with any level of compression. It operates in the YCbCr color space and performs JPEG restoration in two phases using two different autoencoders: the first one restores the luma channel exploiting 2D convolutions; the second one, using the restored luma channel as a guide, restores the chroma channels exploiting 3D convolutions. Extensive experimental results on four widely used benchmark datasets (i.e. LIVE1, BDS500, CLASSIC-5, and Kodak) show that our model outperforms state of the art methods, even those using a different set of weights for each compression quality, in terms of all the evaluation metrics considered (i.e. PSNR, PSNR-B, and SSIM). Furthermore, the proposed model shows a greater robustness than state-of-the-art methods when applied to compression qualities not seen during training.

INDEX TERMS JPEG restoration, deep learning, residual network, autoencoder.

I. INTRODUCTION

Image compression represents a very active research topic due to the high impact of the data in a large number of fields, from image sharing on the web to the most specific applications involving the acquisition of images and transfer to elaboration nodes. Specifically, image compression refers to the task of representing images using the smallest storage space possible.

Compression algorithms play a key role in saving space and bandwidth for the memorization and transfer of large amounts of images. Two different compression paradigms exist: the former is lossless image compression, where the compression rate is limited by the requirement that the original image must be perfectly recovered; the latter, more diffused, is lossy image compression, where higher compression rates are possible at the cost of some distortion in the recovered image. Among the lossy compression algorithms, the most diffused and used is the JPEG compression algorithm.

The JPEG compression algorithm first converts the original RGB image into YCbCr color space and processes the luma and chroma channels separately. It divides the luma channel of an input image into non-overlapping 8×8 blocks and performs the Discrete Cosine Transform (DCT) on each block separately while downsampling the chroma components with a bilinear filter. The DCT coefficients obtained from the luma channel are then quantized based on *quantization tables* and adjusted using the user-selected

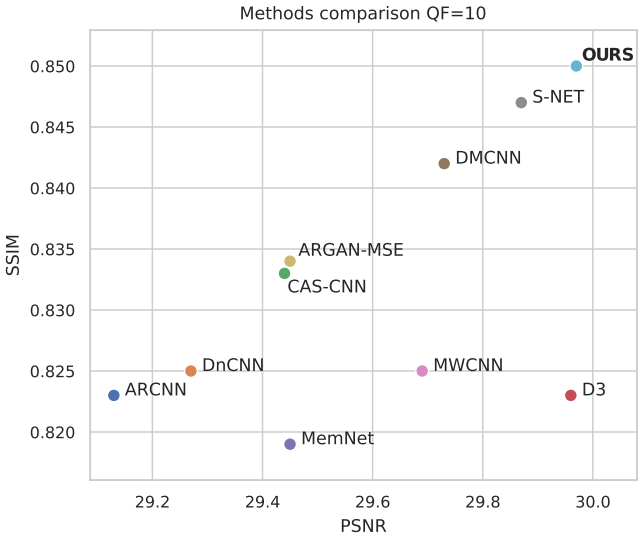

FIGURE 1. PSNR-SSIM comparison of the state-of-the-art-models and our proposed method. For both metrics higher value means better visual results.

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

Quality Factor. The image is then reconstructed from the quantized DCT coefficients by using the inverse DCT. The described JPEG encoding operation introduces three kinds of artifacts in the recovered images, related to the quality factor used for the compression: i) blocking artifacts, which come from the recombination of the 8×8 blocks, that are independently compressed without considering the adjacent blocks; ii) ringing artifacts, which are most visible along the edges and are related to the coarse quantization of the high-frequencies components; iii) blurred low-frequencies areas, which is also related to the compression of the high-frequencies in the DCT domain.

The presence of these kinds of artifacts represents a problem since the general quality of the images is degraded resulting unpleasing for normal users and for generic applications (e.g. projection, print, etc.), or even useless for computer vision applications where the loss of information can be potentially critic for the task [1], [2].

With the purpose of reducing these artifacts, in the last years, a lot of JPEG artifact reduction algorithms have been proposed. These methods include both traditional image processing pipelines [3]–[10] and machine learning approaches [11]–[20], both making great steps in the restoration of corrupted images. However, these methods suffer from two main limits: the first one is that they need to train a different model for each possible Quality Factor (QF), making them not generally applicable to general images downloaded from the web unless the QF used for compression is known; the second one, is that the great majority of methods in the state of the art restores just the luma channel or do not fully exploit the knowledge about the JPEG compression pipeline.

To address these problems we propose a new method for the blind universal restoration of JPEG compressed images, based on machine learning, specifically on convolutional autoencoders. The proposed approach consists of two deep autoencoders respectively used for luma and chroma restoration, that are able to restore images independently from the quality factor used for the compression. The main contributions are the following:

- the design of a method for the blind universal restoration of JPEG compression artifacts that is independent from the QF used;
- the design of a model trainable end-to-end that fully exploits knowledge about JPEG compression pipeline;
- a thorough comparison with the state of the art on three standard datasets at fixed QFs;
- an analysis of the robustness of restoration results at QFs not used for training.

II. RELATED WORKS

The task of JPEG compression artifacts removal has been faced in different ways in the past years. The existing proposed methods can be broadly classified into two groups: traditional image processing methods and learning-based methods.

The first group includes methods based on traditional image processing techniques working both in the spatial and in the frequency domain. For spatial domain processing, different kinds of filters have been proposed, with the intent of restoring specific areas of the images such as edges [3], textures [4], smooth regions [5], etc. Algorithms usually rely on information obtained by the application of the Discrete Cosine Transform (DCT) transform [6]. SA-DCT, proposed by Foi *et* al. [7], attempts to reconstruct an estimate of the signal using the DCT of the original image together with the spatial information contained in the image itself. However, SA-DCT is not capable to reproduce details like sharp edges or complex textures. To overcome this limit different restoration oriented methods have been proposed, like the Regression Tree Fields based method (RTF) [8]. The RTF uses the results of SA-DCT to restore images, taking advantage of a regression tree field model.

Following the success of the application of Deep Convolutional Neural Networks (Deep-CNNs) in image processing tasks, such as image denoising [12] and Single-Image Super-Resolution [21], Deep-CNNs have been applied with success to JPEG compression artifact removal task. The basic idea behind Deep-CNNs is to learn a function to map a set of images from an input distribution, to the desired output one [22]. In the artifact removal case, the objective is to map degraded images into another distribution without the presence of the noise. The trained neural network obtained at the end of the training process represents an approximation of the desired function for the translation of the images from a distribution to another one.

The first attempt with this kind of model has been done by Dong *et al.* [11] who proposed the ARCNN, a model inspired by SRCNN [21], a neural network for Super-Resolution. This first attempt has been followed by DnCNN [12], a CNN for general denoising task that has also been used on JPEG compressed images, and CAS-CNN [13], a model proposed by Cavigelli *et al.*, who presented a much deeper model capable to obtain higher quality images. Wang *et al.* proposed D3 [14], a deep neural network that adopts JPEG-related priors to improve reconstruction quality which obtained an improvement in speed and performances with respect with to the previous models.

In 2018 several new models for JPEG artifact removal have been presented, showing interesting improvements in the quality of the results. Liu *et al.* [16] proposed a Multi-level Wavelet CNN (MWCNN), a model based on the U-Net architecture [23], trained and used for multiple tasks: compression artifact removal, denoising, and super-resolution. Zhang *et al.* [17] developed DMCNN, a Dual-Domain Multi-Scale CNN, which gains higher results quality than the previous works, by using both pixel and frequency (*i.e.* DCT) domain information. Galtieri *et al.* [15] and Yoo *et al.* [20] tried to address the problem of JPEG compressed image restoration by employing a generative adversarial network (GAN) [24] for artifact removal and texture reconstruction. Lastly, two interesting methods have been proposed: S-Net,

by Zheng *et al.* [18], a method based on a "greedy loss architecture" to train deeper models capable to outperform the previous state-of-the-art, and JBCBCR, the most recent method proposed by Chen *et al.* [25], which restores JPEG images in YCbCr space, exploiting the correlation between the information from both luma and chroma components of the images.

III. PROPOSED METHOD

The methods in the state of the art mainly suffer from two limits: the first one is that each machine learning model needs to know the JPEG compression Quality Factor (QF) of each input image to properly restore a compressed image; the second one is that the great majority of them are capable to restore only the luma channel without considering the chroma components. Only the two most recent methods try to restore also the colors of the images: S-Net [18] which works on RGB space, and JBCBR [25] which tries to exploit the distribution of the artifacts coming from the JPEG pipeline, working in YCbCr space.

In this work, we propose a method able to overcome both the aforementioned problems. The first problem has to do with the way the models are trained: all of the previously existing methods make the implicit assumption that the compression quality factor QF that has been used to compress the input images is known at restoration time. In fact, most of the previous models present networks trained on datasets compressed on specific quality factors (the most common being $QF = 10, 20, 30$ and 40). This way of training the models leads to two limits:

- the models are capable to correctly restore only images at a specific QF, with the consequence that specific training for each quality factor is needed;
- the QF used for the compression of the images is needed in order to select a model and correctly restore the images since each model is trained at a specific compression QF. This is usually an unknown information for images coming from unknown sources (e.g. downloaded from the web), thus largely limiting the usability of the models.

In order to overcome the necessity to know the compression quality factor, we train our model on a dataset containing images compressed at different QFs: this will make the model more generic and able to restore images taken in the wild, i.e. without knowing the actual QF used. This objective poses a challenge, since the training of such a quality independent model is much harder than training on a single quality factor: for example, the model has to learn if a strong edge present in the image is a JPEG artifact belonging to an image with a low QF and thus should be corrected, or a real edge belonging to an image with a high QF and therefore should be preserved. Preliminary experiments, in fact, showed that just training a state of the art method with images compressed at different QFs significantly deteriorates the restoration performance with respect to the same method trained for a single QF.

The second problem concerns the way the previous models restore the images: almost all of the previous state-of-the-art methods are trained on the luma channel (*Y* channel of the *YCbCr* space) of the images. This approach is based on the fact that the JPEG compression algorithm applies the DCT to the *Y* channel, introducing ringing and blocking artifacts on the luma channel, while the other *Cb* and *Cr* channels are just sub-sampled the bicubic interpolation. The design and training of a model for the specific restoration of the luma component and its subsequent application for the restoration of the chroma components (as done for example by ARCNN [11]), introduces chromatic aberrations and artifacts in the final result. S-Net [18] and JBCBR [25] are the only methods considering this problem and instead of training a model for the restoration of just the luma component, they work respectively in RGB and YCbCr color spaces for restoring both luma and chroma.

To overcome this second limit and restore also the color information, similarly to Chen *et al.* [25] we exploit the knowledge of how the JPEG compression pipeline works and propose the use of two models for the image restoration in *YCbCr* space: the first model restores the *Y* channel; the second model then uses the result as a *Structure Map* (i.e. a guide) for the restoration of the chroma components. A schematic representation of the proposed method is depicted in Figure 2. The input RGB image is converted into YCbCr color space and the Y channel is separated from the Cb and Cr channels. The Y channel is restored with a dedicated network, and the result is channel-wise concatenated with the original Cb and Cr channels. This stack is processed with a second network that produces as output the restored Cb and Cr channels using the restored Y channels as a guide. The restored Y, Cb, and Cb channels, the former coming from the first network and the latter ones coming from the second network, are channel-wise concatenated and converted from YCbCr to RGB to produce the final output.

A. LUMA AND CHROMA RESTORATION MODEL

The vast majority of learning based methods for JPEG compression artifact removal in the state of the art [11]–[14], [16], [17] focus exclusively on the luma component of the images. Generally, these methods perform the compression artifact removal working on the *Y* channel of the images, after converting them in *YCbCr* color space. These approaches do not take into consideration the chroma aspects of the images, generating results with aberrations in RGB space and low perceptual quality.

, the JPEG compression algorithm, when operating with very low compression quality factors (e.g. $QF \leq 20$) tends to change the colors of the input images in two different ways: hue change and spatial location change. As can be seen in Figure 3, in the compressed version of the *Cb* and *Cr* channels, as expected, the color resolution is reduced and also, for some elements, the color position does not correspond to the one in the original uncompressed image.

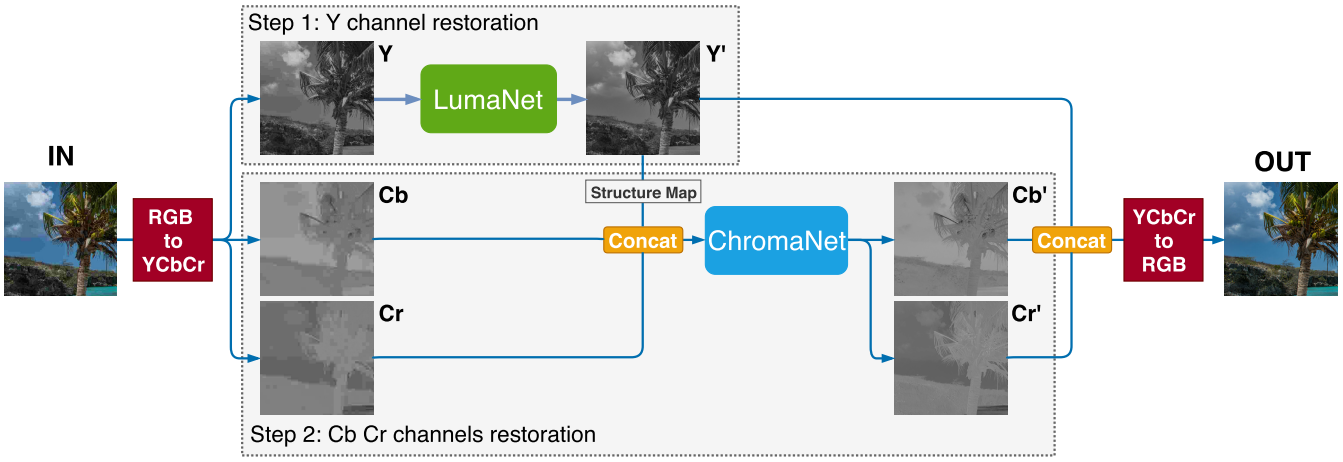

FIGURE 2. Schematic representation of the proposed method: the input image is first converted to *YCbCr* color space. The *Y* channel is restored with the LumaNet and the result *Y'* is concatenated with the original *CbCr* channels to restore *Cb'Cr'* with the ChromaNet. Restored *Y'Cb'Cr'* channels are then converted back to RGB color space.

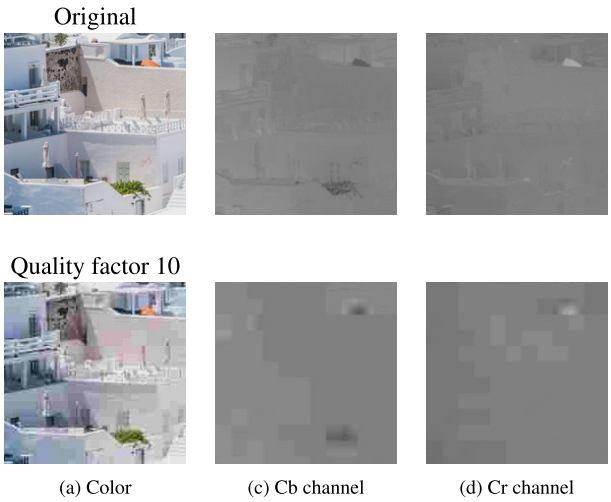

FIGURE 3. Visual example of how the JPEG compression algorithm, when operating with very low compression quality factors, changes the colors of the input images in two different ways: hue change and spatial location change.

In the last years, only two models tried to restore the images considering also the chroma components. These methods are S-Net [18] and JBCBR [25]. While the first one tries to restore the information contained in the images in RGB space, the second one exploits the YCbCr space, the same used for the compression by the JPEG algorithm.

Keeping the above considerations in mind we propose a method for restoring both luma and chroma components of the compressed images (see Figure 2). The method consists of two steps: the first step, after the conversion of the input image into *YCbCr* color space, involves the restoration of the *Y* channel alone, using a first model named LumaNet, and produces *Y'* as output. The second step concatenates *Y'CbCr* along the channel dimension and uses a second model named ChromaNet, to restore the *CbCr* channels. This second step uses *Y'* as a map of the structures present in the image (i.e. a sort of guide) to condition the second network to recover the color hue and contours, and produces *Cb'Cr'* as output. The final output is obtained by concatenating *Y'Cb'Cr'* and converting them back to RGB. Both LumaNet and ChromaNet are two different deep CNN Autoencoders both exploiting a new revisited version of the Residual Blocks [26].

B. DEEP RESIDUAL AUTOENCODER ARCHITECTURE

Autoencoder architectures have been widely used in image processing tasks like image-to-image translation [27], Super-Resolution [28], image inpainting [29] and rain removal [30]. Autoencoders for image processing tasks generally present a structure made by three parts: the encoder, which extracts features from the n-dimensional input (usually one or three channels); a central part, that performs feature processing; and the final decoder, which decodes the processed features into the output image having the desired dimensions. Figure 4 shows a schematic representation of the proposed model, while a more detailed description of its architecture is reported in Table 1.

The encoder, which consists of two convolutions followed by Leaky ReLU activations, is followed by a central part for feature enhancement consisting in a sequence of *Residual-in-Residual Dense Blocks* (RRDB) [31], a modified version of the well known residual blocks originally introduced in the ResNet architecture [26], that have been shown to perform well in other image processing tasks, e.g. image super-resolution [31], [32]. The RRDBs blocks combine multi-level residual learning and dense connection architecture: the RRDBs are designed without the use of the Batch Normalization and the application of the residual learning on different levels. The RRDBs are shown in Figure 5: each RRDB is made of five *Dense Blocks*, which use only convolutions with Leaky ReLUs activation and dense skip connection structures, combined together with other skip connections. Finally, the decoder is designed in a symmetrical way with respect to the encoder part.

The same architecture has been used for both the networks for luma and chroma restoration, but with some differences:

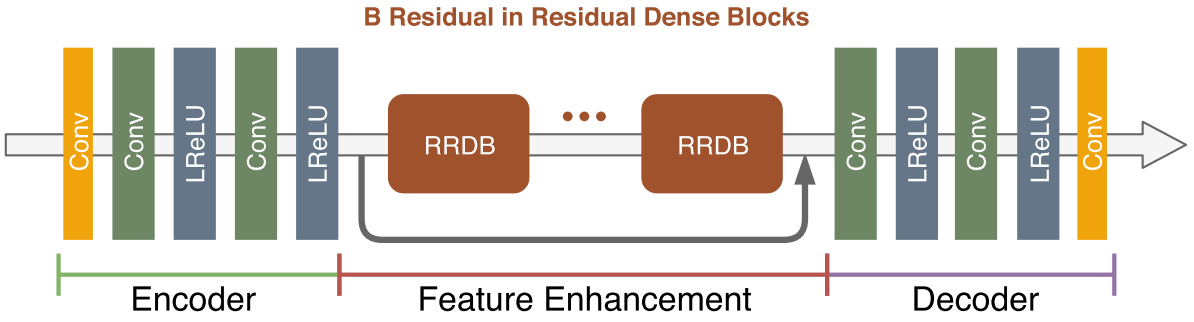

FIGURE 4. Graphical representation of the architecture of the autoencoders used for both the luma and chroma restoration.

TABLE 1. Detailed architecture of the autoencoders used for both the luma and chroma restoration. The number of RRDBs is $B = 5$ for the Y-Net and $B = 3$ for the CbCr-Net.

	Layer	Filter size, Stride, Padding	output channels
	Conv2D	1x1, 1, 0	64
Encoder	Conv2D	3x3, 1, 1	64
	LReLU	-	64
	Conv2D	3x3, 1, 1	64
	LReLU	-	64
	RRDB x B		
Decoder	Conv2D	3x3, 1, 1	64
	LReLU	-	64
	Conv2D	3x3, 1, 1	64
	LReLU	-	64
	Conv2D	1x1, 1, 0	1 / 2
	Tanh	-	1 / 2

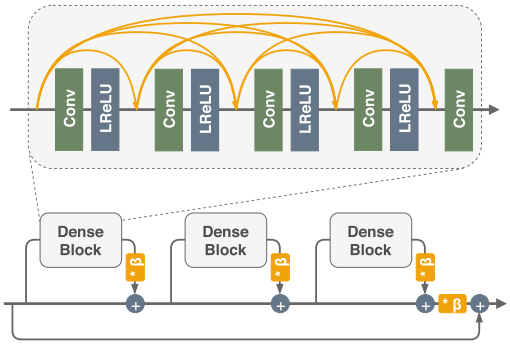

FIGURE 5. Schematic representation of the architecture of the Residual-in-Residual Dense Block (RRDB) [31].

- different depth in terms of number of RRDBs used in the central part;
- different feature extraction from the input in the encoder part.

For the restoration of the luma (*Y* channel), the number of central RRDBs is set to five, while for the *CbCr* restoration the number of RRDB is decreased to three. The second and more important difference is in the first layer of the *CbCr version* of the network, which is a *3-dimensional convolutional layer*. Considering that the input of the *CbCr-Net* is the concatenation (along the channel dimension) of the restored *Y'* channel with the *Cb* and *Cr* channels, we decided to use a 3D convolution to make the model capable to correlate information about color and structures with the use of the same kernels for all the information coming from the three input channels. The output of this second network are the two restored *Cb* and *Cr* channels, which are then concatenated with the restored *Y'* channel, in order to obtain the complete restored image.

In order to improve the quality of the generated results, as well as to make the training process more stable, the proposed architecture includes the following design choices:

- removal of Batch Normalization (BN) layers from the Residual Blocks;
- use of a *residual scaling* parameter in each Residual Block;
- initialization of the model weights using a scaled version of the Kaiming initialization [33].

The removal of the batch normalization layers has been proved, in image Super-Resolution [32] and image deblurring [34] tasks, to increase the performances for the generation of images in terms of quality indexes (PSNR and SSIM [35]). The removal of the BN layers, which improve the stability of the training and the generated image appearance, makes, on the other hand, the training of deep networks more difficult. To solve that issues two solutions have been proved to work well: the so-called residual scaling (in our model set to 0.2), to scale each residual in order to not magnify the input image in a wrong way, and a small weight initialization, obtained by the application of the Kaiming initialization, presented by He *et al.* [33], scaled by a factor 0.1. As can be seen in Figure 5 the residual scaling is applied to the higher level of the residual learning architecture, i.e. on the output of each dense block and at the end of the RRDBs.

IV. EXPERIMENTAL SETUP

The training of the proposed method leads to two different Deep-CNNs respectively for the restoration of the luminance and chroma components of JPEG compressed images at generic quality (i.e. QFs). In order to evaluate the results, our models have been compared with the state of the art in four different experimental setups:

1) *known QF* luminance restoration: comparison with the state-of-the-art methods which work only on the Y channel of the input images;

2) *unknown QF* luminance restoration: comparison to test the ability of the models to restore images at intermediate QFs never seen during training;
3) restoration of areas with high and low details density: evaluation of the performances of the state-of-the-art methods and the proposed one over specific areas of the images, by dividing the images in patches classified on high-to-low frequency (DCT domain) and high-to-low detail density;
4) color restoration: evaluation of the color restoration capability of the model on the images converted in RGB space after the processing.

A. DATASET

The dataset used for training is the DIV2K dataset, a collection of high-quality images (2K resolution), presented during the NTIRE2017 challenge [36] for image restoration tasks. This dataset is made of a total amount of 900 images: 800 are used for training while the remaining 100 are used for validation. The complete dataset contains also 100 images for testing. The ground truth of this last part has not been released after the challenge, and therefore are not used in this paper.

With the purpose of increasing the amount of different texture and pattern to show to the model during training, we have combined the DIV2K dataset with the Flickr2K dataset [37], a collection of 2650 high-quality images (same resolution as the DIV2K) collected from Flickr website.

In order to train the models on different quality factors, for each image in the dataset we have applied 10 different compression levels, corresponding to the quality factors between $QF = 10$ to $QF = 100$, with step 10. The images have been compressed with the MATLAB standard library function. In the training phase of our model, as a preprocessing operation, the compressed images are read and converted into YCbCr space using the Python Scikit-Image library ($v0.14.0$). The compressed version of the training dataset contains 8000 images. The same operation has been applied to the Flickr2K dataset for a total amount of 34k training images.

The evaluation of our model, for the luminance channel restoration, has been done on the LIVE1 [35], Classic-5, BSD500 [38] and Kodak Lossless True Color Image Suite [39], four benchmark datasets widely used for JPEG artifact removal algorithm evaluation. For the evaluation of the behavior of the models with the *unknown compression quality factor* we adopted the SDIVL [40], a dataset proposed for Image Quality Assessment task.

The evaluation of the color channels restoration has been done using the Kodak Lossless True Color Image Suite [39], in the same way that has been done by Chen *et al.* [25].

B. EVALUATION METRICS

The globally adopted metrics for the evaluation of the quality of images in artifact removal tasks are PSNR, PSNR-B [41] (which focus the evaluation on the blocking artifacts) and SSIM [35] indexes. For all of these three measures, a higher value means better results. The PSNR and PSNR-B indexes give information about the quality of the images in terms of noise and perceived quality, with PSNR-B taking into consideration also the blocking artifacts; SSIM index is an indicator of the quality of edges and structures contained in the image. For all the three indexes considered a higher value means that the content and the structures in the reconstructed image are more similar to the ones in the target image.

C. TRAINING DETAILS

All the training phase has been done on an NVIDIA Titan V GPU with 8 GB of memory using PyTorch framework at version 0.4.1. The mini-batch size has been set to 8 and each input image has been cropped to a patch size of 100×100 pixels. During the experiments we tried to train the network with different crop sizes (32×32, 50×50, 100×100 and 400×400), observing how training deeper networks with bigger patch size gives a boost on performances over both PSNR and SSIM indexes.

We also explored the use of different numbers of RRDBs in the model: we observed how with deeper models, using this specific kind of residual blocks, the results got better and better, increasing the PSNR and SSIM values on the validation set. The final structure uses five RRDBs for the Y channel restoration model and three RRDBs for the CbCr model, where each convolution has 64 filters. We found this configuration to be the best one, with respect to the patch size, the amount of RRDBs, the number of filters and the limits due to the memory offered by our board.

We trained the model using Adam optimizer [42] with $\beta_1 = 0.9, \beta_2 = 0.999$, with learning rate initialized at 2×10^{-4} decreased after 200 epochs of training by a factor of 2. The training has been performed using the L1 Loss since allows us to achieve better PSNR results and to make the training more stable.

V. EXPERIMENTAL RESULTS

A. RESTORATION WITH KNOWN COMPRESSION QUALITY FACTOR

We compared our model with the state-of-the-art models ARCNN [11], CAS-CNN [13], D3 [14], and the more recent DMCNN [17], MemNet [19], MWCNN [16], ARGAN [15], S-Net [18] and JBCBR [25].

Since the state-of-the-art methods operate only on the Y channel of the images, in order to make a fair comparison, we used only the result coming from the application of the LumaNet, without any integration of data from the color components. The metrics are evaluated on the Y channel recovered by the first network with the corresponding target images, using the MATLAB standard libraries, over five different compression qualities: 10, 20, 40, 60, 80. For each method, on all the datasets considered, we report the results taken from the corresponding publication, except for ARCNN and MWCNN which provide the source-code, that are then used for the evaluation. Since the training of the

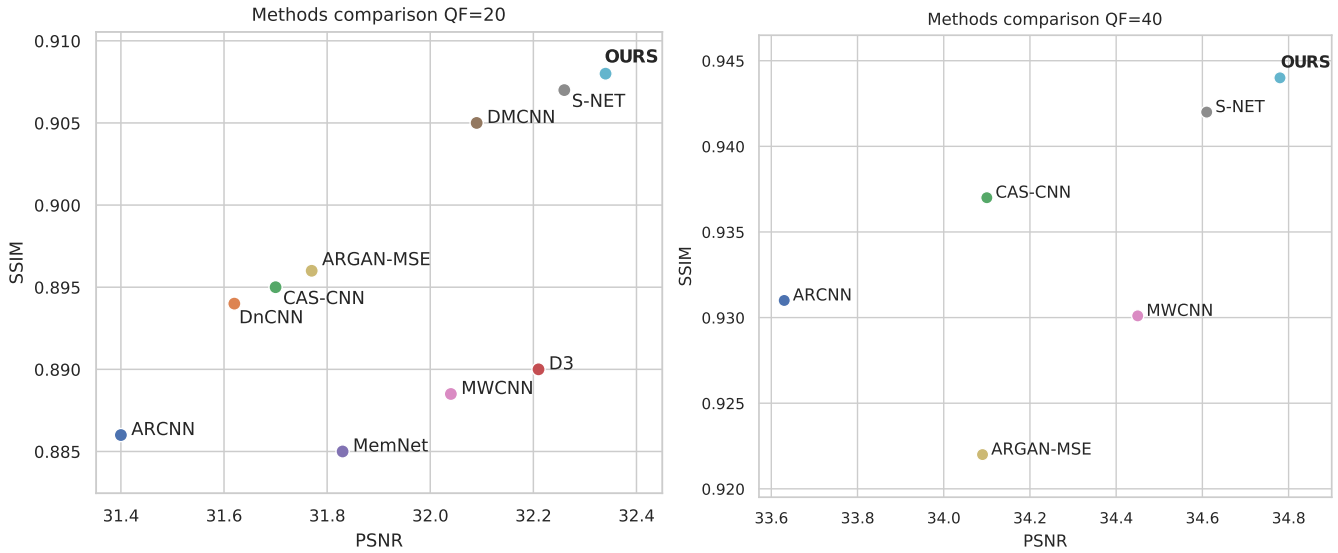

FIGURE 6. PSNR-SSIM comparison of the state-of-the-art-models and our proposed method. For both metrics higher value means better visual results.

TABLE 2. Comparison on test set LIVE1: for the methods in the state of the art a different model is trained for each QF considered. The proposed method uses the same model for all the QFs.

	Quality	ARCNN [11]	DnCNN [12]	CAS-CNN [13]	D3 [14]	DMCNN [17]	MemNet [19]	MWCNN [16]	S-NET [18]	ARGAN-MSE [15]	ARGAN [15]	CED-GT [20]	OURS
PSNR	10	29.13	29.19	29.44	29.96	29.73	29.45	29.37	29.87	29.47	27.65	26.54	**29.98**
	20	31.40	31.59	31.70	32.21	32.09	31.83	31.58	32.26	31.81	29.99	29.33	**32.34**
	40	33.63	33.96	34.10	-	-	-	34.17	34.61	34.17	31.64	-	**34.78**
	60	-	-	35.78	-	-	-	-	-	-	-	-	**36.47**
	80	-	-	38.55	-	-	-	-	-	-	-	-	**39.31**
PSNR-B	10	28.74	-	29.19	29.45	29.55	-	28.85	-	29.13	27.63	26.51	**29.61**
	20	30.69	-	30.88	31.35	31.32	-	30.83	-	31.29	29.69	29.32	**31.76**
	40	33.12	-	33.68	-	-	-	33.33	-	33.42	31.17	-	**33.96**
	60	-	-	35.10	-	-	-	-	-	-	-	-	**35.51**
	80	-	-	37.73	-	-	-	-	-	-	-	-	**38.26**
SSIM	10	0.823	0.812	0.833	0.823	0.842	0.819	0.832	0.847	0.833	0.777	0.767	**0.851**
	20	0.886	0.880	0.895	0.890	0.905	0.885	0.891	0.907	0.897	0.864	0.854	**0.908**
	40	0.931	0.924	0.937	-	-	-	0.936	0.942	0.937	-	0.903	**0.944**
	60	-	-	0.954	-	-	-	-	-	-	-	-	**0.960**
	80	-	-	0.973	-	-	-	-	-	-	-	-	**0.976**

TABLE 3. Comparison on test set BSD500: for the methods in the state of the art a different model is trained for each QF considered. The proposed method uses the same model for all the QFs.

	Quality	ARCNN [11]	DnCNN [12]	CAS-CNN [13]	D3 [14]	DMCNN [17]	MemNet [19]	MWCNN [16]	S-NET [18]	ARGAN-MSE [15]	ARGAN [15]	CED-GT [20]	OURS
PSNR	10	29.10	-	-	-	29.67	-	29.50	29.82	29.05	27.31	26.00	**29.92**
	20	31.25	-	-	-	31.98	-	31.34	32.15	31.23	28.48	28.62	**32.23**
	40	33.55	-	-	-	-	-	33.23	34.45	33.45	30.98	-	**34.61**
PSNR-B	10	28.75	-	-	-	-	-	28.60	-	28.64	27.31	25.97	**29.41**
	20	30.60	-	-	-	-	-	29.84	-	30.49	29.03	28.58	**31.39**
	40	32.80	-	-	-	-	-	31.04	-	32.34	30.16	-	**33.34**
SSIM	10	0.819	-	-	-	0.840	-	0.835	0.844	0.806	0.749	0.731	**0.847**
	20	0.885	-	-	-	0.904	-	0.889	0.905	0.877	0.841	0.825	**0.906**
	40	0.929	-	-	-	-	-	0.928	0.941	0.923	0.884	-	**0.943**

proposed methods leads to a single model that can be used for all the quality factors, we used the same model for the evaluation at all the qualities previously mentioned. All the state-of-the-art methods compared, instead, have a different trained model for each QF considered.

Table 2, 3, 4 and 5 respectively report the comparison on the LIVE1, BSD500, Classic-5, and the Kodak Lossless True Color Image Suite datasets for all the three metrics considered. As can be seen, our model outperforms the state of the art on all the metrics. With the proposed model we obtained improvements with respect to the state-of-the-art methods on both general perceptual quality (PSNR/PSNR-B) and structure reconstruction (SSIM) on the first two datasets. On the third and fourth ones, we obtain improvement in both PSNR-B and SSIM, with comparable results with respect to the best method in terms of PSNR.

Since each index focuses on different aspects of the restoration quality, each index alone is not capable to summarize all the aspects of a good reconstruction. Therefore, we also compare the methods in a graph style-view, reported in Figures 1 and 6 to correlate the two indexes. In order to obtain a more pleasing perceived quality, both the metrics must obtain

TABLE 4. Comparison on test set Classic-5: for the methods in the state of the art a different model is trained for each QF considered. The proposed method uses the same model for all the QFs.

	Quality	ARCNN [11]	DnCNN [12]	CAS-CNN [13]	D3 [14]	DMCNN [17]	MemNet [19]	MWCNN [16]	S-NET [18]	ARGAN-MSE [15]	ARGAN [15]	CED-GT [20]	OURS
PSNR	10	29.04	29.4	-	-	-	29.69	**29.68**	-	-	-	-	29.67
	20	31.16	31.63	-	-	-	31.90	31.78	-	-	-	-	**31.89**
	40	33.34	33.77	-	-	-	-	**34.05**	-	-	-	-	34.04
PSNR-B	10	28.75	-	-	-	-	-	29.06	-	-	-	-	**29.35**
	20	30.6	-	-	-	-	-	30.95	-	-	-	-	**31.43**
	40	32.8	-	-	-	-	-	33.20	-	-	-	-	**33.33**
SSIM	10	0.811	0.803	-	-	-	0.811	0.828	-	-	-	-	**0.829**
	20	0.869	0.861	-	-	-	0.866	0.878	-	-	-	-	**0.882**
	40	0.91	0.9	-	-	-	-	0.916	-	-	-	-	**0.917**

TABLE 5. Comparison on test set Kodak Lossless True Color Image Suite: for the methods in the state of the art a different model is trained for each QF considered. The proposed method uses the same model for all the QFs. The values marked with the symbol (*) are taken from [25] while the other ones are obtained using the codes officially released by the corresponding authors, and the evaluation code from [11].

	Quality	D2SD [10]	ARCNN [11]	DnCNN [12]	MemNet [19]	MWCNN [16]	JBCBR [25]	OURS
				Y				
PSNR	10	30.28*	30.01 / 30.56*	30.75*	30.96*	30.82 / **31.19***	31.03	31.10
	20	32.45*	32.31 / 32.78*	33.09*	33.29*	33.10 / **33.47***	33.31	33.44
PSNR-B	10	-	29.88	-	-	30.63	30.82	**30.93**
	20	-	32.06	-	-	32.81	32.99	**33.17**
SSIM	10	-	0.818	-	-	0.836	0.846	**0.847**
	20	-	0.881	-	-	0.896	0.902	**0.904**

high values. It is easy from this kind of view to see how the proposed method outperforms the current state-of-the-art models even if a single model is used for all the QFs.

B. RESTORATION WITH UNKNOWN COMPRESSION QUALITY FACTOR

Another kind of evaluation has been done about the capability of the models to recover images at compression quality factors never seen during training. In most of the real use-cases, the JPEG compression quality factor previously applied to an image is not known: it is then important that a model can recover the images without this prior information. On the other hand, if we are able at least to estimate the compression quality factor of the input compressed image, following the previous approaches we should train new models for each specific quality factor needed, or use the model trained for the closest QF to the desired one.

We compare our model with the two state-of-the-art models for which the code is available (i.e. ARCNN and MWCNN) in a specific selection of cases. Since previous models have been trained on specific quality factors, and our model has been instead trained over quality factor from 10 to 100 in steps of 10, without the use of images with QFs in between, we decided to test the model robustness on ''never seen'' artifacts. In order to perform the evaluation coherently, for the state-of-the-art algorithm we used the pretrained models for the nearest quality factor, for example, if the input image has been compressed with $QF = 17$ we used the models trained for $QF = 20$. The evaluation has been done only on the Luminance channel restores only with the LumaNet, in the same way, that has been done for the known QFs. For this evaluation we adopted the SDIVL dataset: for each image of the testset, we applied all of the compression factors in the interval $5 - 25$. The evaluation is done in the same way it has been done in the previous section, by extracting Y channel and measuring PSNR, PSNR-B, and SSIM indexes.

In Figure 7 are shown the results of the models on the SDIVL with all the quality factors compression. As can be seen in those graphs our model shows a more stable behavior: the model is capable to restore images at different QFs with more coherent and smooth behavior in relation to the increase of the QF, in comparison with the other methods. Moreover, the previous state-of-the-art models have difficulties to restore images at quality factors distant from the trained one. It is particularly interesting to see how the other models have difficulties to restore images at higher qualities with respect to the QF used in training, in terms of structures in the images (Figure V-B), due to the more complex textures never seen by the models during the training phase.

C. HIGH AND LOW FREQUENCY AREAS RESTORATION

In order to better understand if the proposed method performs better than approaches in the state of the art only on certain image types, we conduct a further experiment: we divide the images from LIVE1 testset, compressed at $QF = 10$, into 64×64 patches and classify each of them into five categories. The categories are obtained by equally diving the patches into five bins with respect to both frequency and detail density. Patch frequency is computed as the weighted average of the 2D Fourier Transform normalized magnitude. Patch detail density is computed as the 2D average of the result of the Canny edge detection. The results for the considered evaluation metrics over the five categories of the frequency and detail density are respectively reported in Table 6 and 7. From the results reported it is possible to notice that the proposed method consistently outperforms the state of the art on all the frequency and detail density categories.

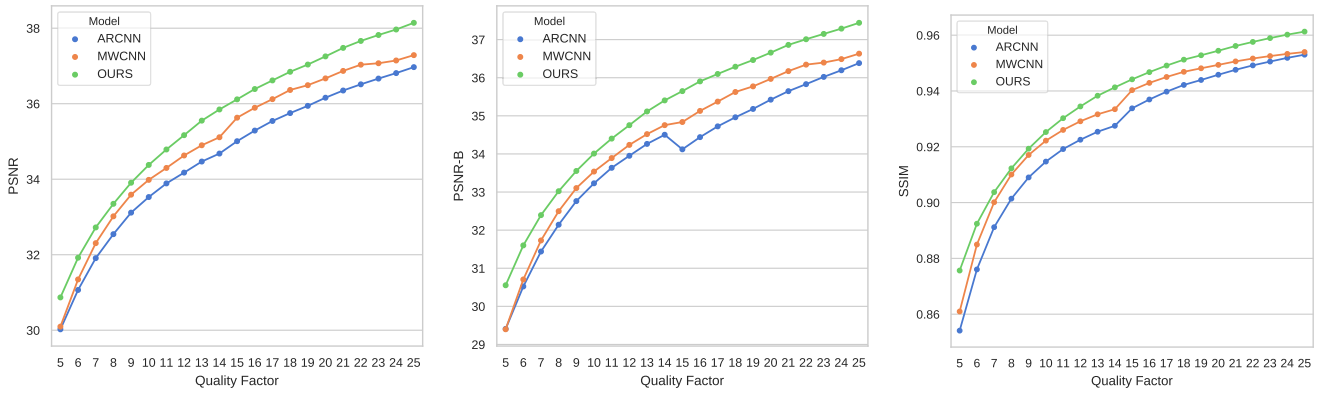

FIGURE 7. Comparison on QFs not seen during training. For ARCNN and MWCNN the models trained for QF=10 and QF=20 are tested on QF in the range [5, 25]. The proposed model is trained for QF in the range [10, 100] with steps of 10, and is tested on the same intermediate QFs not seen in training.

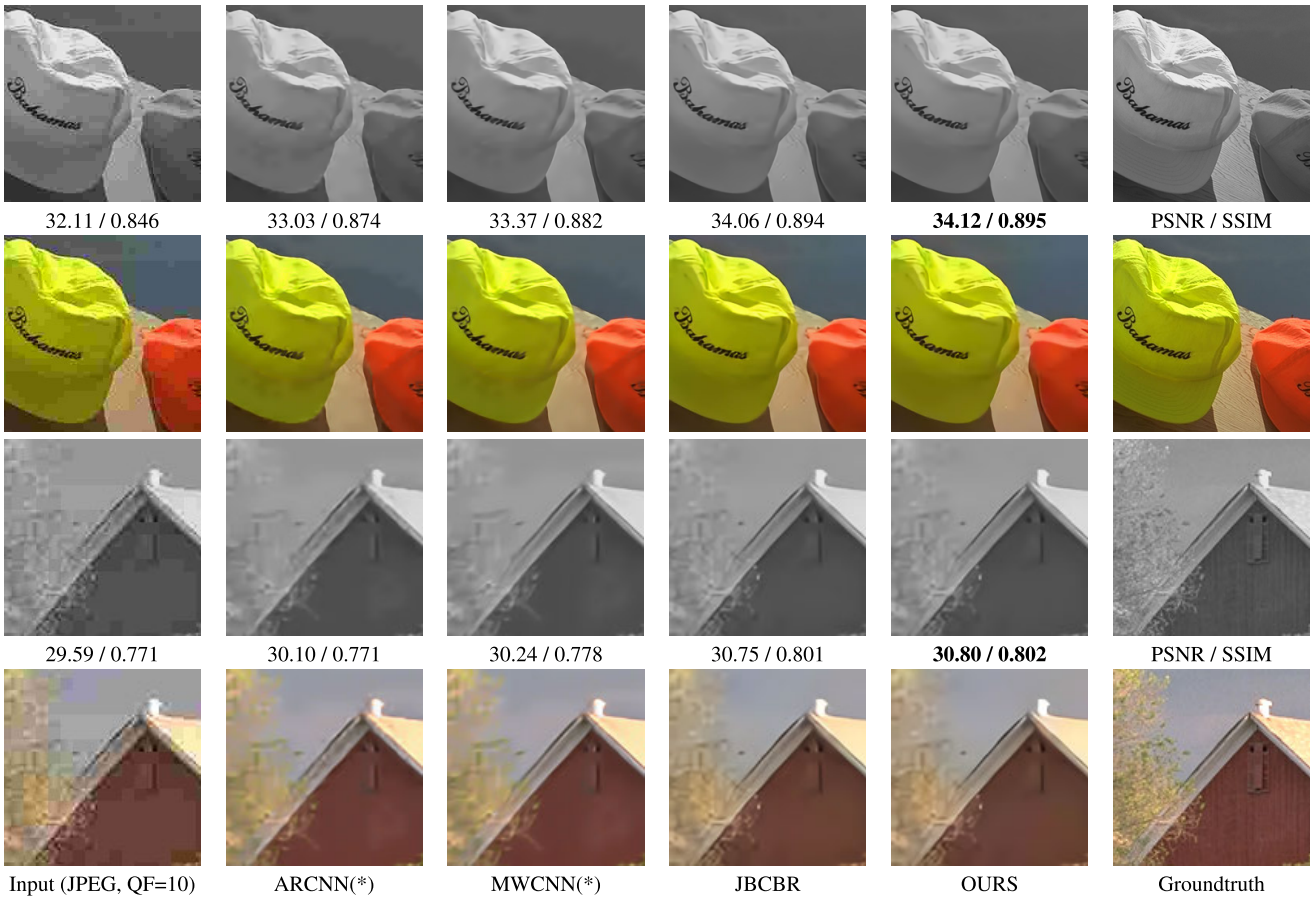

FIGURE 8. Visual comparison of image restoration result. The first and third lines show the Luma channel (Y) restored by the models with the associated PSNR and SSIM values, computed on the whole image; the second and forth lines show the RGB colored version. For the models that can only recover the Y channel (identified by the * symbol), the Cb and Cr channels are taken directly from the original high quality corresponding ground truths crops.

D. COLOR RESTORATION

The final evaluation is focused on the color restoration capability of the models. The comparison has been done evaluating the Cb and Cr channels of the recovered images, in the same way that has been done by Chen *et al.* [25]: we compared the chroma component of the restored images from the Kodak Lossless True Color Image Suite, with $QF = 10$ and $QF = 20$. From the results reported in Table 8 in terms of PSNR it is possible to see that the proposed model obtains better results than the other methods. This is remarkable since, analogously to what has been done for the previous experiments, the compared methods trained a different

TABLE 6. Comparison on test set LIVE1 by subdividing the image patches on the basis of the frequency content in five classes from high to low.

	ARCNN [11]			MWCNN [16]			OURS		
Frequency	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
high	27.53	27.26	0.782	27.61	27.24	0.792	**28.18**	**27.88**	**0.807**
medium-high	25.00	24.66	0.685	25.24	24.67	0.700	**25.64**	**25.18**	**0.729**
medium	24.61	24.27	0.734	24.50	23.82	0.740	**25.37**	**24.91**	**0.773**
medium-low	25.92	25.49	0.794	25.91	25.24	0.803	**26.73**	**26.21**	**0.827**
low	27.08	25.93	0.840	26.72	25.27	0.849	**27.81**	**26.52**	**0.864**

TABLE 7. Comparison on test set LIVE1 by subdividing the image patches on the basis of the detail density in five classes from low to high.

	ARCNN [11]			MWCNN [16]			OURS		
Edges frequency	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
high	23.20	22.94	0.667	23.42	22.82	0.683	**23.87**	**23.45**	**0.716**
medium-high	24.69	24.39	0.721	24.91	24.31	0.735	**25.42**	**25.02**	**0.763**
medium	25.68	25.22	0.758	25.94	25.26	0.772	**26.41**	**25.86**	**0.794**
medium-low	26.83	26.12	0.805	27.01	25.95	0.817	**27.47**	**26.61**	**0.832**
low	29.17	28.22	0.884	27.45	26.28	0.888	**29.97**	**28.99**	**0.897**

TABLE 8. Color restoration comparison on test set Kodak Lossless True Color Image Suite. Evaluation of Cb and Cr channels restoration in terms of PSNR.

Quality	JPEG	SA-DCT [7]	JBF [9]	EJBF [9]	JBCBR [25]	OURS
			Cb			
10	36.14	37.83	37.38	37.39	39.16	**39.30**
20	39.02	40.69	40.34	40.45	41.99	**42.22**
			Cr			
10	36.00	37.57	37.16	37.20	38.92	**39.04**
20	38.99	40.47	40.01	40.24	41.64	**41.89**

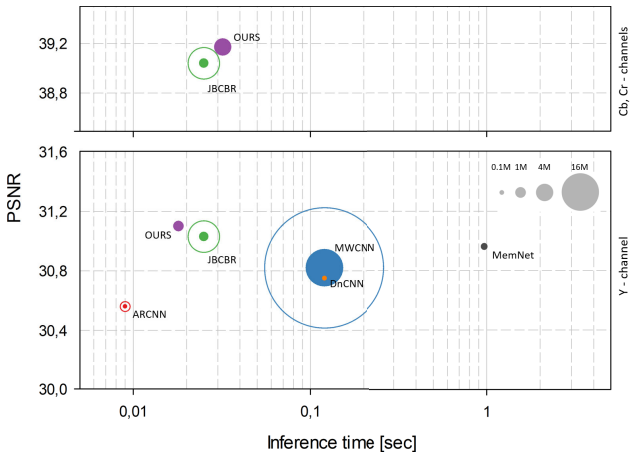

FIGURE 9. Inference time for a 512 × 512 × 3 image on a NVIDIA Titan V GPU. In the top plot the average PSNR on the Cb and Cr channels is reported, in the bottom plot the PSNR on the Y channel is reported.

method for each QF considered, while the proposed method uses a single model for all QFs. A visual comparison of color image restoration results is reported in Figure 8. In particular, it is possible to see how the methods that are trained to recover full color images obtain much better visual results even on the luma channel alone. In order to not perform an unfair visual comparison, for the methods designed to recover just the luma channel, the Cb and Cr channels are taken from the original uncompressed image.

VI. MODEL COMPLEXITY

In this section, we perform a comparison of model complexity. In particular, we compare the inference time for a single 512 × 512 × 3 on an NVIDIA Titan V GPU, the PSNR score and the model size in terms of learnable parameters. The results are reported in Figure 9. They are divided into two plots: the bottom one reports the comparison for the restoration of the Y-channel, the top one reports the comparison for the restoration of the Cb and Cr channels. For some methods, two sizes are reported: the full circle represents the size for a single model (i.e. trained for a single QF) while the empty circle represents the size of ten models, simulating the fact that at test time we need to have more models to cover all the possible QFs. The plots show that our solution compares favorably with respect to the state of the art on all the aspects considered, showing also a very good tradeoff between PSNR and model size.

VII. CONCLUSION

In this paper we proposed a deep residual autoencoder exploiting Residual-in-Residual Dense Blocks (RRDB) to remove artifacts in JPEG compressed images. The proposed method is blind and universal, i.e. it is independent from the QF used. The proposed model operates in the YCbCr color space and performs a two-phase restoration of JPEG artifacts: in the former phase, a first autoencoder exploiting 2D convolutions is used to restore the luma channel of the input image. In the latter phase, the restored luma is stacked along the channel dimension with the chroma channels of the input image; then, a second autoencoder employing 3D convolutions uses the restored luma channel as a guide to restore the chroma channels.

The main contributions of this paper are: i) the design of a blind universal method for the restoration of JPEG compression artifact that is independent from the QF used; ii) the design of a model trainable end-to-end that fully exploits knowledge about JPEG compression pipeline; iii) a thorough comparison with the state of the art on four standard datasets at fixed QFs; iv) an analysis of robustness of restoration results at QFs not used for training.

Extensive experimental results on four widely used benchmark datasets (i.e. LIVE1, BDS500, CLASSIC-5, and Kodak) show that our model is able to outperform the state of the art with respect to all the evaluation metrics considered (i.e. PSNR, PSNR-B, and SSIM). This result is remarkable since the approaches in the state of the art use a different set of weights for each compression quality, while the proposed model uses the same weights for all of them, making it applicable to images in the wild where the QF used for compression is unknown. Furthermore, the proposed model shows greater robustness than state-of-the-art methods when applied to compression qualities not seen during training. Since preliminary experiments with the same architecture proposed showed good results for the restoration of other artifacts (i.e. noise removal, in the CVPRW NTIRE2019 challenge [43]), as future work we plan to investigate its extension to other single and multiple distortions [44]. To this end, techniques that are able to better interpret and understand what the model has learned, such as what has been done in the framework of image classification [45], [46], should be studied to be applied also in the image processing domain.

ACKNOWLEDGMENT

The authors would like to thank the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

REFERENCES

[1] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *Proc. 8th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Jun. 2016, pp. 1–6.

[2] S. Bianco, L. Celona, and R. Schettini, "Robust smile detection using convolutional neural networks," *J. Electron. Imag.*, vol. 25, no. 6, Nov. 2016, Art. no. 063002.

[3] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[4] H. C. Reeve, III, and J. S. Lim, "Reduction of blocking effects in image coding," *Opt. Eng.*, vol. 23, no. 1, Feb. 1984, Art. no. 230134.

[5] C. Wang, J. Zhou, and S. Liu, "Adaptive non-local means filter for image deblocking," *Signal Process., Image Commun.*, vol. 28, no. 5, pp. 522–530, May 2013.

[6] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. 100, no. 1, pp. 90–93, Jan. 1974.

[7] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality deblocking of compressed color images," in *Proc. 14th Eur. Signal Process. Conf.*, Sep. 2006, pp. 1–5.

[8] J. Jancsary, S. Nowozin, and C. Rother, "Loss-specific training of non-parametric image restoration models: A new state of the art," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 112–125.

[9] N. Wada, M. Kazui, and M. Haseyama, "[Paper] extended joint bilateral filter for the reduction of color bleeding in compressed image and video," *ITE Trans. Media Technol. Appl.*, vol. 3, no. 1, pp. 95–106, 2015.

[10] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven soft decoding of compressed images in dual transform-pixel domain," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1649–1659, Apr. 2016.

[11] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 576–584.

[12] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[13] L. Cavigelli, P. Hager, and L. Benini, "CAS-CNN: A deep convolutional neural network for image compression artifact suppression," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 752–759.

[14] 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. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2764–2772.

[15] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 2131–2145, Aug. 2019.

[16] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 773–782.

[17] X. Zhang, W. Yang, Y. Hu, and J. Liu, "Dmcnn: Dual-domain multi-scale convolutional neural network for compression artifacts removal," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 390–394.

[18] B. Zheng, R. Sun, X. Tian, and Y. Chen, "S-net: A scalable convolutional neural network for jpeg compression artifact reduction," *J. Electron. Imag.*, vol. 27, no. 4, 2018, Art. no. 043037.

[19] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4539–4547.

[20] N. Kwak, J. Yoo, and S.-H. Lee, "Image restoration by estimating frequency distribution of local patches," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6684–6692.

[21] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Computer-Assisted Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[25] H. Chen, X. He, C. An, and T. Q. Nguyen, "Deep wide-activated residual network based joint blocking and color bleeding artifacts reduction for 4:2:0 JPEG-compressed images," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 79–83, Jan. 2019.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[28] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao, "Coupled deep autoencoder for single image super-resolution," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 27–37, Jan. 2017.

[29] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.

[30] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2482–2491.

[31] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, Sep. 2018, pp. 63–79.

[32] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, vol. 1, no. 2, p. 4.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[34] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 1, no. 2, p. 3.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[36] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.

[37] R. Timofte *et al.*, "NTIRE 2017 challenge on single image super-resolution: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1110–1121.
[38] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
[39] *Kodak Lossless True Color Image Suite*. Accessed: Apr. 3, 2020. [Online]. Available: http://r0k.us/graphics/kodak/index.html
[40] S. Corchs, F. Gasparini, and R. Schettini, "No reference image quality classification for JPEG-distorted images," *Digit. Signal Process.*, vol. 30, pp. 86–100, Jul. 2014.
[41] C. Yim and A. C. Bovik, "Quality assessment of deblocked images," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 88–98, Jan. 2011.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[43] A. Abdelhamed, R. Timofte, and M. S. Brown, "Ntire 2019 challenge on real image denoising: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019.
[44] S. Corchs and F. Gasparini, "A multidistortion database for image quality," in *Proc. Int. Workshop Comput. Color Imag.* Cham, Switzerland: Springer, 2017, pp. 95–104.
[45] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2018.
[46] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6541–6549.

SIMONE ZINI received the B.Sc. and M.Sc. degrees in computer science from the Department of Informatics, Systems, and Communication (DISCo), University of Milano–Bicocca, Italy, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the DISCo. His current research interests concern machine learning, image enhancement, and computational photography.

SIMONE BIANCO received the B.Sc. and M.Sc. degrees in mathematics and the Ph.D. degree in computer science from the University of Milano–Bicocca, Italy, in 2003, 2006, and 2010, respectively. He is currently an Associate Professor of computer science with the Department of Informatics, Systems, and Communication, University of Milano–Bicocca. His current research interests include computer vision, machine learning, optimization algorithms, and color imaging.

RAIMONDO SCHETTINI has been associated with the Italian National Research Council, since 1987, where he has led the Color Imaging Laboratory, from 1990 to 2002. He is currently a Full Professor with the University of Milano–Bicocca, Italy. He is also the Head of the Imaging and Vision Laboratory. He has been the team leader of several research projects. He has authored or coauthored over 300 refereed articles and holds several patents on color reproduction, image processing, analysis, and classification. He is a Fellow of the International Association of Pattern Recognition for his contributions to pattern recognition research and color image analysis.