Discernible Compressed Images via Deep Perception Consistency

Zhaohui Yang\(^1\), Yunhe Wang\(^2\), Chao Xu\(^3\), Chang Xu\(^1\)

\(^1\) Key Lab of Machine Perception (MOE), Peking University \(^2\) Huawei Noah’s Ark Lab
\(^3\) School of Computer Science, Faculty of Engineering, The University of Sydney

zhao@pku.edu.cn, yunhe.wang@huawei.com
xuchao@cis.pku.edu.cn, c.xu@sydney.edu.au

Abstract

Image compression, as one of the fundamental low-level image processing tasks, is very essential for computer vision. Conventional image compression methods tend to obtain compressed images by minimizing their appearance discrepancy with the corresponding original images, but pay little attention to their efficacy in downstream perception tasks, e.g., image recognition and object detection. In contrast, this paper aims to produce compressed images by pursuing both appearance and perception consistency. Based on the encoder-decoder framework, we propose using a pre-trained CNN to extract features of original and compressed images. In addition, the maximum mean discrepancy (MMD) is employed to minimize the difference between feature distributions. The resulting compression network can generate images with high image quality and preserve the consistent perception in the feature domain, so that these images can be well recognized by pre-trained machine learning models. Experiments on benchmarks demonstrate the superiority of the proposed algorithm over comparison methods.

1 Introduction

Recently, more and more computer vision (CV) tasks such as image recognition \cite{Simonyan2015, He2015}, visual segmentation \cite{Long2015}, object detection \cite{Girshick2014, Ren2015}, and face verification \cite{Sun2014}, are well addressed by deep neural networks, which benefits from the large amount of accessible training data and computational power of GPUs. Besides these high-level CV tasks, a lot of low-level CV tasks have been enhanced by neural networks, e.g., image denoising and inpainting \cite{Burger2012, Zhang2017, Xie2012}, single image super-resolution \cite{Dong2016}, and image and video compression \cite{Theis2017, Toderici2017, Rippel2017, Hamper2018}. This paper studies the image compression problem, a fundamental approach for saving storage and transmission consumptions, which represents images with low-bit data and reconstructs them with high image quality. Traditional methods are mainly based on the time-frequency domain transform (e.g., JPEG \cite{Wallace1992} and JPEG 2000 \cite{Skodras2001}), which makes compressed images distorted with blocking artifacts or noises. Since convolutional networks have shown extraordinary performance on image denoising and inpainting \cite{Burger2012, Xie2017, Dong2015, Zhang2017} proposed using CNN to remove blocks on JPEG compressed images in order to enhance the compression performance. Moreover, \cite{Toderici2015} utilized an encoder-decoder network to implement the compressing task with fixed input size. \cite{Toderici2017} further extended the encoder-decoder network to a general model which supports images with arbitrary sizes. \cite{Sun2018} utilized the recursive dilated network to establish the image compression system. \cite{Li2018} proposed to compress images by exploiting the importance map to achieve higher compression rates.

Although these methods obtained promising performance to reduce the storage of digital images, there is an important issue should be considered. In practice, a number of digital images will be taken and stored in electronic devices (e.g., mobile phones, security camera), and a large proportion of them will be recognized or post processed using pre-trained machine models for person identification, object recognition and detection, etc. Therefore, we do not expect that compressed images cannot be accurately recognized after compressing. Fig. 1 shows a toy experiment, we directly remove a proportion of frequency coefficients of the original image in the DCT domain and recognize compressed image using a pre-trained ResNet \cite{He2015}. Although there is only a very small appearance difference between the original image and the compressed image, some underlying structure and textual changes will affect the calculation of the subsequent neural network. The network recognizes some compressed Pepper images as Hamper, though there is no hurdle for use to recognize them by eyes. On the other side, although we could retrain existing models (e.g., classification or detection) for fitting these compressed images, the time consumption is not tolerable. Therefore, an image compression method for generating compressed images with perception consistency in the feature domain is urgently required.

To address the aforementioned problems, this paper develops a novel image compression framework which simultaneously executes image compression and image recognition.
tasks. In specific, an encoder-decoder network is used for generating compressed data and reconstructed images, and a pre-trained CNN is adopted to perceive the difference of images after and before compression. By jointly optimizing these two objectives, the proposed method can produce compressed images with low storage which can also be accurately perceived as usual by pre-trained CNN models. To the best of our knowledge, this is the first time to simultaneously investigate the physical compression and visual perception of images using deep learning methods. Experiments conducted on benchmark datasets demonstrate the superiority of the proposed algorithm over the state-of-the-art methods for compressing digital images.

2 Percepcion-consistent Image Compression

The encoder-decoder network receives and outputs images and represents them with activations of a number of hidden neurons, which is naturally suitable for implementing the image compression task [Theis et al., 2017; Toderici et al., 2015; Toderici et al., 2017].

2.1 Encoder-decoder for Image Compression

Generally, the loss function of an encoder-decoder based image compression network can be written as

$$\min_{\theta_1, \theta_2} \frac{1}{n} \sum_{i=1}^{n} ||D(\theta_2, E(\theta_1, x^i)) - x^i||^2,$$  \hspace{1cm} (1)

where $E(\cdot)$ is the encoder network with parameter $\theta_1$ for compressing the given image $x^i$, $n$ is the number of images, and $D(\cdot)$ is the decoder network with parameter $\theta_2$ for recovering the compressed data to the original image. To simplify the expression, the compressed data $c$ and decoded image $y$ are denoted as

$$c^i = E(\theta_1, x^i), \quad y^i = D(\theta_2, c^i),$$  \hspace{1cm} (2)

respectively. Since the encoder network $E$ consists of a series of transforms such as convolution, pooling, binary (or quantization) and pruning, the decoded image $y^i$ resulted from conventional JPEG algorithm [Wallace, 1992] often has some distortions such as blocks and artifacts. Therefore, a feasible way for enhancing the image quality of $y^i$ is to use another operation or model for refining the decoded image $y^i$:

$$\min_{\hat{y}} \frac{1}{n} \sum_{i=1}^{n} (||\hat{y}^i - y^i||^2 + \lambda R(\hat{y}^i)),$$  \hspace{1cm} (3)

where $\hat{y}^i$ is the recovered image and $R(\cdot)$ can be some conventional regularizations for natural images such as total variation (TV) norm, $\ell_1$ norm, etc. These techniques have been widely used in conventional image processing methods [Gu et al., 2014; Dong et al., 2011]. In contrast, neural networks of strong capacity can generate clearer images than those of traditional methods. Additional layers can be easily added after $D(\cdot)$ for refining $y^i$, so that the above function can be absorbed by Fcn. 1 to form an end-to-end model learning framework.

An ideal image compression algorithm should not only concentrate on the low storage of generated images, but also have to retain the downstream performance of tasks such as image recognition and object detection, etc. Therefore, a neural network with parameter $\theta_3$ is introduced for supervising the generated image:

$$\min_{\theta_1, \theta_2, \theta_3} \frac{1}{n} \sum_{i=1}^{n} (||y^i - x^i||^2 + \lambda L(y^i, \theta_3)),$$  \hspace{1cm} (4)

$$\text{s.t.} \quad y^i = D(\theta_2, E(\theta_1, x^i)),$$

where $L(\cdot)$ can be various based on different applications, e.g., cross-entropy loss, regression loss.

There are a number of off-the-shelf visual models (e.g., ResNet [He et al., 2015], VGGNet [Simonyan and Zisserman, 2015], RCNN [ Girshick et al., 2014]) well trained on large-scale datasets (e.g., ISLVRC [Russakovsky et al., 2015] and COCO [Lin et al., 2014]), and completely retraining them is a serious waste of resources and is time consuming. In addition, most existing deep learning models only support input images with fixed sizes (e.g., $224 \times 224$), while a complete image compression system should be used for processing images with different sizes.

Hence, we propose to use a pre-trained neural network $F(\cdot)$ as a “perceptron” to process of original and compressed images simultaneously. Therefore, we formulate a novel image compression method giving consideration to both appearance and perception consistency:

$$\min_{\theta_1, \theta_2} \frac{1}{n} \sum_{i=1}^{n} (||y^i - x^i||^2 + \lambda |F(y^i) - F(x^i)||^2),$$  \hspace{1cm} (5)

$$\text{s.t.} \quad y^i = D(\theta_2, E(\theta_1, x^i)),$$

where $\lambda$ is the trade-off parameter, and parameters in $F(\cdot)$ are fixed. The diagram of the proposed network is shown in Fig. 2. Since the size of decoded image can be various according to different sizes of input image, $F(\cdot)$ cannot be the
same as the original pre-trained network. In practice, $\mathcal{F}(\cdot)$ is a pre-trained CNN after discarding the last several layers, which can generate features with different dimensionalities given different images. Although Fcn. 5 does not explicitly optimize the recognition performance, considerable convolution filters in $\mathcal{F}(\cdot)$ trained over a number of images can be beneficial for perceiving. Therefore, minimizing Fcn. 5 will encourage the perception consistency between compressed image $y$ and its corresponding original image.

### 2.2 Feature Distribution Optimization

A novel image compression model was proposed in Fcn. 5, which introduces a new module $\mathcal{F}(\cdot)$ for extracting visual features of original and compressed images. Since there are considerable neurons in a well-designed neural network, $\mathcal{F}(\cdot)$ will convert input images into high-dimensional (e.g., 512) features, and it is very hard to directly minimize differences between features of these images. Therefore, we propose to use another measurement to supervise the compression task, i.e., maximum mean discrepancy (MMD [Sejdinovic et al., 2013; Long et al., 2015b]), which is used for describing differences of two distributions by mapping sample data in kernel spaces.

Suppose we are given an image dataset with $n$ images, and two sets of image features $\mathcal{X} = \{\mathcal{F}(x^i)\}_{i=1}^n$ sampled from distribution $p$ and $\mathcal{Y} = \{\mathcal{F}(y^i)\}_{i=1}^n$ sampled from distribution $q$, where $x^i$ and $y^i$ are the $i$-th original image and the $i$-th compressed image, respectively. The squared formulation of MMD distance between $p$ and $q$ is defined as:

$$\mathcal{L}_{MMD}(\mathcal{X}, \mathcal{Y}) \triangleq \frac{1}{n} \left| \sum_{i=1}^{n} \psi(\mathcal{F}(x^i)) - \sum_{i=1}^{n} \psi(\mathcal{F}(y^i)) \right|^2, \quad (6)$$

where $\psi(\cdot)$ is an explicit mapping function. It is clear that feature distributions of original and compressed images are exactly the same iff $\mathcal{L}_{MMD} = 0$, i.e., $p = q$ [Sejdinovic et al., 2013]. The above function can be further expanded with the kernel trick:

$$\mathcal{L}_{MMD}(\mathcal{X}, \mathcal{Y}) = \frac{1}{n^2} \left[ \sum_{i=1}^{n} \sum_{i'=1}^{n} k(\mathcal{F}(x^i), \mathcal{F}(x^{i'})) + \sum_{i=1}^{n} \sum_{i'=1}^{n} k(\mathcal{F}(y^i), \mathcal{F}(y^{i'})) - \sum_{i=1}^{n} \sum_{i'=1}^{n} k(\mathcal{F}(x^i), \mathcal{F}(y^{i'})) \right],$$

where $k(\cdot, \cdot)$ is a kernel function for projecting given data into a higher or infinite dimensional space, which can be set as linear kernel, Gaussian kernel, etc. Since each kernel has its own functionality for measuring distributions of data, it is very hard to determine which one is the best in practice without time consuming cross-validation. Therefore, we borrow the strategy in [Long et al., 2015b] to use a set of kernels for projecting features:

$$\left\{ k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geq 0, \forall u \right\},$$

where $m$ is the number of kernels, $\beta_u$ is the coefficient of the $u$-th kernel which can be optimized iteratively. Therefore, we reformulate Fcn. 5 as

$$\mathcal{L}_{Comp}(\theta_1, \theta_2) = \frac{1}{n^2} \sum_{i=1}^{n} \left| y^i - x^i \right|^2 + \frac{\lambda}{n^2} \sum_{i=1}^{n} \left| \mathcal{F}(y^i) - \mathcal{F}(x^i) \right|^2 + \gamma \mathcal{L}_{MMD}(\mathcal{X}, \mathcal{Y}), \quad (7)$$

s.t. $y^i = \mathcal{D}(\theta_2, \mathcal{E}(\theta_1, x^i))$, $\mathcal{X} = \{\mathcal{F}(x^i)\}_{i=1}^{n}$, $\mathcal{Y} = \{\mathcal{F}(y^i)\}_{i=1}^{n}$, where $\gamma$ is the weight parameter for the MMD loss. By simultaneously optimizing the compression loss and the perception loss, we can obtain a model which generates compressed images of the consistent perception with original images for a series of downstream tasks such as image recognition and segmentation, etc. Alg. 1 summarizes the mini-batch strategy of the proposed method for learning image compression network. In addition, the pre-trained network $\mathcal{F}(\cdot)$ will be dropped after the training process.
Discussion. The proposed method includes the image compression task and the visual perception task. Wherein, a pre-trained neural network is utilized for extracting features of original and compressed images, which is similar to two categories of works, i.e., transfer learning [Long et al., 2015b] and teacher-student learning paradigm [Hinton et al., 2015; Romero et al., 2014], which also utilize a pre-trained model for inheriting useful information for helping the training process. The main difference is that we do not train any new parameters for the visual recognition task, and parameters in the pre-trained network are fixed, which is used as a powerful regularization for supervising the learning of the encoder-decoder network therefore improves the compression performance.

3 Experiments

Baseline Model: There are a number of CNN based image compression models [Theis and Bethge, 2015; Theis et al., 2017; Luo et al., 2016; Sun et al., 2018; Toderici et al., 2017; Baig et al., 2017; Mentzer et al., 2018], each of which has its own pros and cons. We selected [Toderici et al., 2017] as the baseline model, which utilizes a recurrent neural network (RNN) for compressing images for the following two reasons: 1) the RNN based encoder-decoder network can provide compressed images with different compression rates in each iteration; 2) this model allows the size of the input image to be arbitrary, which is more flexible than comparison methods with fixed input size.

Training Setup: To have a fair comparison, we follow the setting in [Toderici et al., 2017] to conduct the image compression experiment. Each image in the training dataset was decomposed into $32 \times 32$ non-overlapping patches and we train 200 epochs in total. The architecture of RNN used in following experiments is the same as that described in [Toderici et al., 2017], and networks were trained using the PyTorch toolbox. As for the subsequent CNN $F(\cdot)$, we used the ResNet-18 network [He et al., 2015] which shows excellent performance on visual recognition tasks (e.g., a 89.1% top-5 accuracy and a 69.8% top-1 accuracy on the ILSVRC 2012 dataset with 1000 different labels). All parameters in this network were pre-trained on the ImageNet dataset and will be fixed in following experiments. Note that, $F(\cdot)$ used here consists of the first 14 convolutional layers in ResNet-18, since the size of the input image is much smaller than that in the original ResNet-18. In specific, for a given input image size of $32 \times 32$, $F(\cdot)$ outputs a 512-dimensional feature.

Evaluation Metrics: Peak signal to noise ratio (PSNR) and structural similarity (SSIM) are two widely used criteria for evaluating image quality by comparing original images with compressed images. However, the PSNR only measures the mean square error between the compressed image and its original one, and SSIM ignores image differences in different scales. According to [Toderici et al., 2017], besides PSNR and SSIM, we also employ the multi-scale structural similarity (MS-SSIM) [Wang et al., 2003] for evaluating the performance of the proposed image compression algorithm. The MS-SSIM is applied on each RGB channel and we averaged them as the evaluation result. In addition, MS-SSIM values are between 0 and 1. For a given compression rate, a higher MS-SSIM value implies better compression performance.

Impact of parameters: The proposed image compression method as detailed in Alg. 1 has several important parameters: the weight parameter $\lambda$ for reducing difference between features of original and compressed images similar, $\gamma$ is further used for making features distributions of these images similar which was equal to 1, $u$ was set as 8, and $k$ was set as a serious of Gaussian kernels, as analyzed in [Long et al., 2015b]. $b$ is the batch size, which was set as 192 as suggested in [Toderici et al., 2017]. It is clear that $\lambda$ is the most important parameter for balancing appearance difference evaluated by human eyes and perception difference assessed by machines. Therefore, we first tested the impact of this parameter on the ILSVRC-2012 dataset using the ResNet-18 network as shown Table 1. Wherein, each model is trained on the COCO dataset [Lin et al., 2014] is as the same as that of the baseline FRIC (Full Resolution Image Compression) method [Toderici et al., 2017], and we then employ them on the validation dataset of ILSVRC 2012, respectively. This dataset consists of 50,000 images with different scales and ground-truth labels. All images were first compressed by the proposed DIC and several state-of-the-art methods, respectively, and then recognized by ResNet-18. In addition, the bpp value of original RGB images is 24, and we can achieve a $48\times$ compression rate when bpp $= 0.5$, e.g., the file size of a $1MB$ image after compression is about $20KB$, which is totally

$\begin{align*}
\text{Table 1: Recognition results on the ILSVRC 2012 dataset with different hyper-parameters $\lambda$.}
\end{align*}$

| Method  | ResNet-18 | MS-SSIM | PSNR  |
|---------|-----------|---------|-------|
|         | top-1     | top-5   |       |
| Original| 69.8%     | 89.1%   | 1.000 |
| FRIC    | 63.5%     | 85.0%   | 0.921 |
| $\lambda = 10^{-4}$ | 63.3% | 84.8% | 0.917 |
| $\lambda = 10^{-6}$ | 64.0% | 85.4% | 0.925 |
| $\lambda = 10^{-8}$ | 63.9% | 85.3% | 0.924 |

$\begin{align*}
\text{Discussion. The proposed method includes the image compression task and the visual perception task. Wherein, a pre-trained neural network is utilized for extracting features of original and compressed images, which is similar to two categories of works, i.e., transfer learning [Long et al., 2015b] and teacher-student learning paradigm [Hinton et al., 2015; Romero et al., 2014], which also utilize a pre-trained model for inheriting useful information for helping the training process. The main difference is that we do not train any new parameters for the visual recognition task, and parameters in the pre-trained network are fixed, which is used as a powerful regularization for supervising the learning of the encoder-decoder network therefore improves the compression performance.}
\end{align*}$

$\begin{align*}
\text{Evaluation Metrics: Peak signal to noise ratio (PSNR) and structural similarity (SSIM) are two widely used criteria for evaluating image quality by comparing original images with compressed images. However, the PSNR only measures the mean square error between the compressed image and its original one, and SSIM ignores image differences in different scales. According to [Toderici et al., 2017], besides PSNR and SSIM, we also employ the multi-scale structural similarity (MS-SSIM) [Wang et al., 2003] for evaluating the performance of the proposed image compression algorithm. The MS-SSIM is applied on each RGB channel and we averaged them as the evaluation result. In addition, MS-SSIM values are between 0 and 1. For a given compression rate, a higher MS-SSIM value implies better compression performance.}
\end{align*}$

$\begin{align*}
\text{Impact of parameters: The proposed image compression method as detailed in Alg. 1 has several important parameters: the weight parameter $\lambda$ for reducing difference between features of original and compressed images similar, $\gamma$ is further used for making features distributions of these images similar which was equal to 1, $u$ was set as 8, and $k$ was set as a serious of Gaussian kernels, as analyzed in [Long et al., 2015b]. $b$ is the batch size, which was set as 192 as suggested in [Toderici et al., 2017]. It is clear that $\lambda$ is the most important parameter for balancing appearance difference evaluated by human eyes and perception difference assessed by machines. Therefore, we first tested the impact of this parameter on the ILSVRC-2012 dataset using the ResNet-18 network as shown Table 1. Wherein, each model is trained on the COCO dataset [Lin et al., 2014] is as the same as that of the baseline FRIC (Full Resolution Image Compression) method [Toderici et al., 2017], and we then employ them on the validation dataset of ILSVRC 2012, respectively. This dataset consists of 50,000 images with different scales and ground-truth labels. All images were first compressed by the proposed DIC and several state-of-the-art methods, respectively, and then recognized by ResNet-18. In addition, the bpp value of original RGB images is 24, and we can achieve a $48\times$ compression rate when bpp $= 0.5$, e.g., the file size of a $1MB$ image after compression is about $20KB$, which is totally}
\end{align*}$
Table 2: Recognition results on the ILSVRC 2012 dataset, bpp = 0.5.

| Method                  | ResNet-18     | ResNet-50     | PSNR         |
|-------------------------|---------------|---------------|--------------|
|                         | top-1 acc.    | top-5 acc.    | top-1 acc.   | top-5 acc.   |
| Original                | 69.8%         | 89.1%         | 76.2%        | 92.9%        |
| JPEG                    | 62.3%         | 84.2%         | 68.6%        | 88.0%        |
| FRIC [Toderici et al., 2017] | 63.5%         | 85.0%         | 69.2%        | 89.0%        |
| DIC w/o MMD             | 64.0%         | 85.4%         | 69.6%        | 89.1%        |
| DIC w/ MMD              | 64.3%         | 85.5%         | 69.8%        | 89.2%        |

Comparision Experiments: After investigating the trade-off between the performance of image compression and visual recognition, we then compare the proposed method with state-of-the-art compression methods on the ILSVRC 2012 dataset to verify its effectiveness. All images were first compressed by the proposed DIC (Discernible Images Compression) and several state-of-the-art methods, respectively, and then recognized by ResNet-18. Since the pre-trained network $\mathcal{F}(\cdot)$ in Fcn. 7 used for extracting features of images before and after compression is part of the ResNet-18, we also employed the ResNet-50 network to further recognize these images to verify the generalization ability of the proposed method. This network achieves a 89.2% top-5 accuracy and a 69.8% top-1 accuracy on the ILSVRC 2012 dataset.

Compression results are detailed in Table 2, where both ResNet-18 and ResNet-50 were pre-trained on the ILSVRC 2012 dataset, and the experiment here aims to investigate how the image compression algorithm affects the subsequent machine learning tasks. Note that, lower bpp values are frequently discussed in many works for obtaining higher compression rates [Li et al., 2018; Rippel and Bourdev, 2017], but compressed images with bpp values lower than 0.5 have obvious distortions on compressed image, where results of standard JPEG algorithm are also provided for an explicit comparison.

It can be found in Table 2, compressed images generated by all methods downgrade the performance of the subsequent recognition task. It is worth mentioning that, FRIC achieved relatively higher results, since the recurrent network can recover the compressed data iteratively. In contrast, the proposed method can provide compressed images with the highest recognition accuracy on both ResNet-18 and ResNet-50 networks. In addition, some recognition results of com-
pressed images are illustrated in Fig. 3-4. Since the proposed image compression method can preserve the perception consistency, images compressed by the proposed method can be recognized, while predictions on FRIC are biased, e.g., a Trombone was recognized as a Barbetshop as shown in Fig. 4.

Moreover, we further removed the MMD loss, i.e., the last term in Fcn. 7, and re-trained a new model for compressing images to test the impact of the introduced feature distribution regularization. This model was denoted as DIC without MMD (i.e., $\gamma = 0$ in Fcn. 7), and recognition results of compressed images using both ResNet-18 and ResNet-50 were reported in Table 2. The proposed DIC after removing the MMD regularization has an obvious accuracy decline, e.g., its top-1 acc. of ResNet-18 is about 0.3% lower than that of the whole DIC method. Since $F(\cdot)$ converts input images into 512-dimensional features, and it is very hard to directly minimize differences between these high-dimensional features of original and compressed images. The MMD loss thus can provide a more powerful regularization for obtaining better results.

Object Detection after Compressing: Besides the image classification experiment, we further verify the effectiveness of the proposed discernible compressed image generation method on a more complex computer vision application, i.e., object detection. In practice, the deep neural network receives an input image and then outputs locations and labels of each object in the image. Therefore, a slight distortion on the input image could severely damage the prediction results. We selected the SSD (Single Shot MultiBox Detector [Liu et al., 2016]) trained on the VOC 0712 (Visual Object Classes [Everingham et al., 2010]) as the baseline model to conduct the following experiments.

Table 3 reports the detailed mAP (mean Average Precision) values of original and compressed images using different methods, and averaged MS-SSIM results on the VOC 2007 validation set. It is clear that the proposed DIC maintains a higher detection performance. In addition, Fig. 7 illustrates some object detection results of original images and compressed images by exploiting the conventional FRIC and the proposed method. It is obvious that the pre-trained SSD model can still detect and recognize objects in compressed images using the proposed DIC algorithm, but cannot accurately recognize those images compressed by exploiting conventional methods.

4 Conclusions
This paper investigates perception consistency for image compression by embedding a pre-trained neural network into the existing encoder-decoder network. Beyond directly minimizing the distortion between original images and compressed ones generated by the decoder network, we take the perception loss on features into consideration. Compared to state-of-the-art methods, we can generate compressed images of higher image quality by retaining their perception results simultaneously. Experiments on benchmark datasets show that the proposed DIC method can not only produce clearer images of lower storage, but also has limited influence on the downstream visual recognition tasks. The proposed image compression scheme creates a bridge to connect human and machine perceptions. It can be easily deployed for other applications such as denoising, super-resolution, and tracking.
References

[Baig et al., 2017] Mohammad Haris Baig, Vladlen Koltun, and Lorenzo Torresani. Learning to inpaint for image compression. In NIPS, pages 1246–1255, 2017.

[Burger et al., 2012] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In CVPR, 2012.

[Dong et al., 2011] Weisheng Dong, Lei Zhang, and Guangming Shi. Centralized sparse representation for image restoration. In ICCV, pages 1259–1266, 2011.

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

[Dong et al., 2016] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. IEEE TPAMI, 38(2):295–307, 2016.

[Everingham et al., 2010] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. IJCV, 88(2):303–338, 2010.

[Girshick et al., 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[Gu et al., 2014] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In CVPR, 2014.

[He et al., 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.

[Hinton et al., 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[Li et al., 2018] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. CVPR, 2018.

[Lin et al., 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014.

[Liu et al., 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In ECCV, 2016.

[Long et al., 2015a] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[Long et al., 2015b] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In ICML, 2015.

[Luo et al., 2016] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, Xiaoou Tang, et al. Face model compression by distilling knowledge from neurons. In AAAI, pages 3560–3566, 2016.

[Mentzer et al., 2018] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In CVPR, 2018.

[Ren et al., 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[Rippel and Bourdev, 2017] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. arXiv preprint arXiv:1705.05823, 2017.

[Romero et al., 2014] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.

[Russakovsky et al., 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. IJCV, 115(3):211–252, 2015.

[Sejdinovic et al., 2013] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. The Annals of Statistics, pages 2263–2291, 2013.

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

[Skodras et al., 2001] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. IEEE Signal processing magazine, 18(5):36–58, 2001.

[Sun et al., 2014] Yi Sun, Yuheng Xia, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In NIPS, 2014.

[Sun et al., 2018] Liyan Sun, Zhiwen Fan, Yue Huang, Xinghao Ding, and John Paisley. Compressed sensing mri using a recursive dilated network. In AAAI, 2018.

[Theis and Bethge, 2015] Lucas Theis and Matthias Bethge. Generative image modeling using spatial lstms. In NIPS, 2015.

[Theis et al., 2017] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with convolution autoencoders. arXiv preprint arXiv:1703.00395, 2017.

[Toderici et al., 2015] George Toderici, Sean M O’Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. arXiv preprint arXiv:1511.06085, 2015.

[Toderici et al., 2017] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In CVPR, pages 5435–5443, 2017.

[Wallace, 1992] Gregory K Wallace. The jpeg still picture compression standard. Consumer Electronics, IEEE Transactions on, 38(1):xvii–xxiv, 1992.

[Wang et al., 2003] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on, 2003.

[Xie et al., 2012] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In NIPS, 2012.

[Zhang et al., 2017] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE TIP, 2017.