A Reference Resource Based End-to-End Image Compression Scheme

Wenbin Yin[1(✉)], Xiaopeng Fan[1], Yunhui Shi[2], and Wangmeng Zuo[1]

[1] Harbin Institute of Technology, Harbin, Heilongjiang, China
{ywb, fxp}@hit.edu.cn, cswmzuo@gmail.com
[2] Beijing University of Technology, Beijing, China
syhzm@bjut.edu.cn

Abstract. Deep learning and convolutional neural networks have achieved a great success in computer vision and image processing, especially in low-level vision problems such as image compression. Recently, some end-to-end image compression methods have been proposed leading to a new direction of image compression. In this paper, we propose an end-to-end reference resource based image compression scheme to exploit the strong correlations with external similar images. In the proposed scheme, the side information is generated from highly correlated images in the reference resource. The features of side information can conceptually guide the compression process and assist the reconstruction process. The important map is employed to guide the allocation of local bit rate of the residual features. The proposed compression scheme is formulated as a rate distortion optimization problem in an end-to-end manner which is solved by ADAM algorithm. Experimental results prove that the proposed compression framework greatly outperforms several image compression frameworks.

Keywords: Convolutional neural networks · Reference resource
Image compression · Rate distortion optimization

1 Introduction

In recent years, image compression attracts increasing interest in image processing and computer vision area due to its potential applications in many vision systems. The typical image encoding standards [1] (such as JPEG and JPEG2000) generally rely on handcrafted image transformation and separate optimization on codecs, and thus are suboptimal for image compression. Moreover, while the compression ratio increases, the bits per pixel (BPP) decreases as a result of the use of bigger quantization steps, which will cause the decoded image to have some annoying visual artifacts.

This work is supported in part by the National Science Foundation (NSFC) of China under Grants 61672066 and 61472018, and Beijing municipal science and technology commission (Z171100 004417023).

R. Hong et al. (Eds.): PCM 2018, LNCS 11164, pp. 534–544, 2018.
https://doi.org/10.1007/978-3-030-00776-8_49

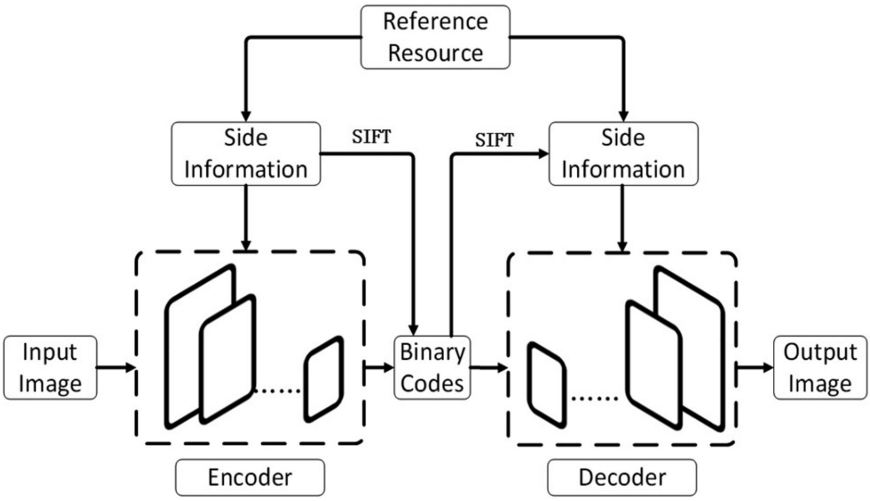

Fig. 1. Framework of the proposed method

For the existing image standards, the codecs actually are separately optimized. In the encoding stage, they first perform a linear transform to an image. Quantization and lossless entropy coding are then utilized to minimize the compression rate. For example, JPEG applies discrete cosine transform (DCT) on 8×8 image patches, quantizes the frequency components and compresses the quantized codes with a variant of Huffman encoding. JPEG 2000 uses a multi-scale orthogonal wavelet decomposition to transform an image, and encodes the quantized codes with the Embedded Block Coding with Optimal Truncation. In the decoding stage, the decoding algorithm and inverse transform are designed to minimize the distortion. In contrast, CNN based methods treat image compression as a joint rate distortion optimization problem, where both nonlinear encoder and decoder are jointly trained in an end-to-end manner.

Recently, deep convolutional networks have achieved great success in computer vision task [2–6]. As to image compression, convolutional networks are also expected to be more powerful than JPEG and JPEG 2000 by considering the following reasons. First, for image encoding and decoding, flexible nonlinear analysis and synthesis transformations can be easily achieved by stacking several convolutional layers. Second, it allows jointly optimizing the nonlinear encoder and decoder in an end-to-end manner. For the lossy image compression, Toderici et al. [7] propose a general framework for variable-rate image compression and a novel architecture based on convolutional and deconvolutional LSTM recurrent networks. Li et al. [8] propose a content weighted compression method with the importance map of the input image. Theis et al. [9] propose compressive autoencoders, which uses a smooth approximation of the discrete of the rounding function and upper-bound the discrete entropy rate loss for continuous relaxation. Balle et al. [10] make use of a generalized divisive normalization (GDN) for joint nonlinearity and replace rounding quantization with additive uniform noise for continuous relaxation.

Imagine a reference resource that collects a huge number of images. When you randomly take a picture, you can often find some highly correlated images in the reference resource that were taken at the same location with different viewpoints and angles, focal lengths, and illuminations. However, the CNN based image compression methods make it hard to utilize external images for compression even if highly correlated image can be found in the reference resource.

Motivated by the excellent performance of convolutional neural networks in image processing. We propose a reference resource based image compression framework. The proposed scheme extracts highly correlated images in the reference resource as side information. The features of side information can guide the compression process and assist the reconstruction process. The importance map is employed to guide the allocation of local bit rate of the residual features. The proposed compression scheme is formulated as a rate distortion optimization problem in an end-to-end manner.

The rest of the paper is organized as follows. Section 2 describes the encoding and decoding process of the proposed scheme. The performance of our scheme is shown in Sect. 3, followed by concluding remarks in Sect. 4.

2 Proposed Method

Our framework contains four components: reference resource (RS) based convolutional encoder, importance map network, binarizer and RR based convolutional decoder. Figure 1 shows the architecture of the proposed framework. Given an input image, the encoder first describes the input image by SIFT descriptors. SIFT descriptors are extracted from the original images and are used to retrieve near and partial duplicate images in the reference resource and identify corresponding patches. The SIFT descriptors are compressed and transmitted to decoder while it could find similar images in the reference resource by utilizing the SIFT descriptors. Then the encoder defines a nonlinear analysis transform by stacking convolutional layers which utilizes side information generated from similar images in the reference resource. The importance map network takes the intermediate feature maps as the input, and yields a content-weighted importance map. Then the binary code is trimmed based on the mask generated from the importance map. The decoder defines a nonlinear synthesis transform to produce decoding result.

2.1 Generation of Side Information

Given an input image, we need to find out its near and partial-duplicate images in the reference resource. The SIFT feature (key-point and descriptor) is one of the most robust and distinctive point features. The SIFT keypoint gains invariance to scale and rotation by exploiting scale-space extrema and the local dominant orientation. The SIFT descriptor assemble a 4×4 array of 8 gradient orientation histograms around the keypoint, making it robust to image variations induced by both photometric and geometric changes.

In the proposed scheme, we extract the side information of current input image at both encoder side and decoder side. After we obtain the SIFT descriptors of current

encoding image, the encoder first transmits them to the decoder with conventional SIFT compression method [11]. In an image, the region of a SIFT descriptor with a large-scale index often partially or completely covers the regions of some SIFT descriptors with small-scale indices. As mentioned in [12], we bundle them as a group. One image often has many groups. Every group is represented by a set of visual words and their geometric relationship in the image. Decoded SIFT feature vectors are quantized to visual words and organized into groups too. Every group is matched with all groups in the reference resource. The number of matched visual words and their geometric relationship score the matching result. This score is assigned to the image that contains the group. After all groups are matched, the image with highest sum scores will be selected as to generate side information of encoding image. Then we will find the corresponding patch from the side information for each encoding patch (Fig. 2).

Fig. 2. The input image and its corresponding similar image in the reference resource

2.2 Convolutional Encoder and Decoder

Both the encoder and decoder in our framework are fully convolution networks and can be trained by back-propagation. The encoder consists of nine convolutional layers. Each three of them are treated as a group. We further remove the batch normalization operations from the last two convolutional layers in one group. The encoder utilizes side information to guide the compression process. The most different point between conventional CNN based image-coding methods and the proposed framework is that we adopt two inputs (input image and corresponding side information) for convolutional networks. There are two networks in the proposed method: feature exaction network and encoding network. The feature exaction network has the similar structure with [8]. The feature exaction networks at both encoder and decoder side are jointly trained to produce feature maps of side information in each level. The feature exaction network can be expressed as

$$F_y^l(y) = w_y^l(y) + B_y^l \tag{1}$$

Where y is the side information, l stands for layer number, B and w represents the biases and the mapping to be learn. The encoding network of proposed method can be formulated as

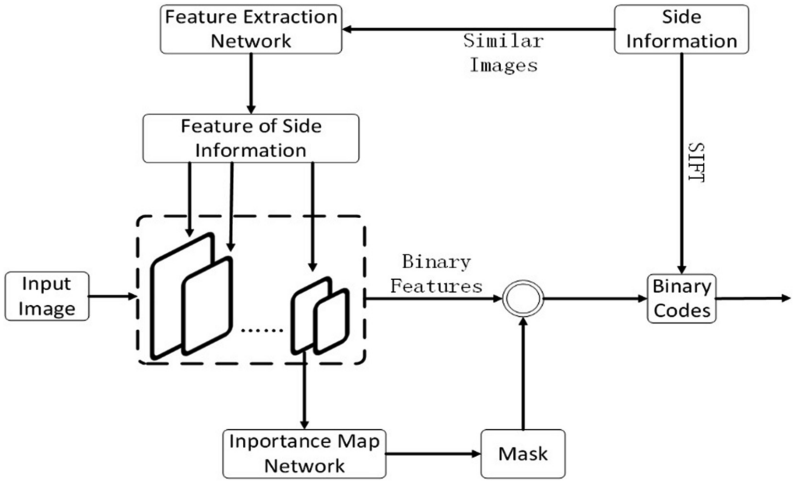

Fig. 3. Encode process of the proposed scheme

$$F_x^l(x) = w_x^l(A \circ F_y^l + F_x^{l-1}(x)) + B_x^l \tag{2}$$

where x is the input image, $\circ$ denotes the element wise multiplication operation and A represents the weights between input image features and side information features (Fig. 3).

In the encoding network, the input image is first convolved with 128 filters with size 8×8 and stride four and followed by one CNN group. The feature maps are then convolved with 256 filters with 4×4 and stride two and followed by two CNN groups to output the intermediate feature maps. Finally, the intermediate feature maps are convolved with n filters with size 1×1 to yield the encoder output. It should be noted that we set n = 64 for low compression rate models with less than 0.5 bpp and n = 128 otherwise. The network architecture of decoder is symmetric to that of the encoder without the importance map network (Fig. 4).

2.3 Binarizer and Importance Map

Since sigmoid nonlinearity is adopted in the last convolutional layer, the encoder output should be in the range of $[0, 1]$. e_{ijk} denotes an element in output of encoder. The binarizer is defined as

$$B(e_{ijk}) = \begin{cases} 1, & \text{if} \quad e_{ijk} > 0.5 \\ 0, & \text{if} \quad e_{ijk} \leq 0.5 \end{cases} \tag{3}$$

However, the gradient of the binarizer function is zero almost everywhere except that it is infinite when $e_{ijk} = 0.5$. In the back-propagation process, the gradient is computed layer by layer by utilizing the chain rule in a backward manner. This will make any layer before the binarizer never be updated during training.

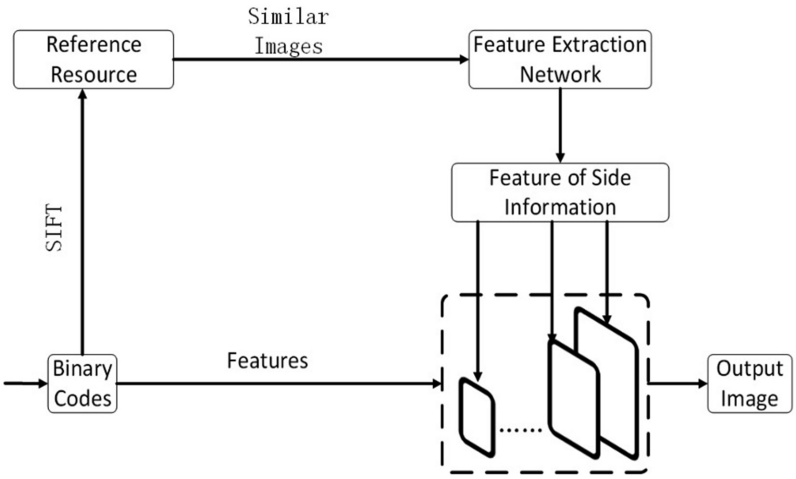

Fig. 4. Decode process of the proposed method

To solve this problem, we adopt a proxy function $\widetilde{B}(e_{ijk})$ to approximate $B(e_{ijk})$. While $B(e_{ijk})$ is still used in forward propagation and $\widetilde{B}(e_{ijk})$ is used in back-propagation. The proxy function is based on the straight though estimator on gradient [2] and is defined as

$$\tilde{B}(e_{ijk}) = \begin{cases} 1, & \text{if } e_{ijk} > 1 \\ e_{ijk}, & \text{if } 0 \le e_{ijk} \le 1 \\ 0, & \text{if } e_{ijk} \le 0 \end{cases} \tag{4}$$

And the gradient of $\widetilde{B}(e_{ijk})$ can be easily calculated by

$$\tilde{B}'(e_{ijk}) = \begin{cases} 1, & \text{if } 0 \le e_{ijk} \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

In general, the attentions in compressing different parts of an image should be different. The smooth regions in an image should be easier to be compressed than those with objects or rich textures. Thus, fewer bits should be allocated to the smooth region while more bits should be allocated to the region with more information. Moreover, when the whole code length for an image is limited, such allocation can also be used for rate control.

The importance map is a feature map with only one channel, and its size should be same with the encoder output. The value of importance map is in the range of (0, 1). An importance map network is deployed to learn the importance map from an input image. It takes the intermediate feature maps from the last residual block of the encoder as input and use a network to produce the importance map.

Denote by h × w the size of the importance map p, n the number of feature maps of the encoder output and $n \times h \times w$ the size of importance mask m. Given an element p_{ij} in p, the quantizer is defined as

$$Q(p_{ij}) = \begin{cases} l-1, & \text{if } \frac{l-1}{L} \le p_{ij} < \frac{l}{L},\ l = 1, \ldots, L \\ L, & \text{if } p_{ij} = 1 \end{cases} \tag{6}$$

where L is the importance levels and $(n \mod L) = 0$. With $Q(p)$, the importance mask can be obtained by

$$m_{kij} = \begin{cases} 1, & \text{if } k \le \frac{n}{L} Q(p_{ij}) \\ 0, & \text{else} \end{cases} \tag{7}$$

The final coding result of the image can be represented as $c = M(p) \circ B(e)$, where $\circ$ denotes the element wise multiplication operation. Finally, in back-propagation, the importance map m can be equivalently rewritten as a function of p

$$m_{kij} = \begin{cases} 1, & \text{if } \left\lceil \frac{kL}{n} \right\rceil < L p_{ij} + 1 \\ 0, & \text{else} \end{cases} \tag{8}$$

where $\lceil . \rceil$ is the ceiling function. Analogous to binarizer, we also adopt a straight-though estimator of the gradient.

2.4 Entropy Encoder

Due to the fact that no entropy constraint is included, the code generated by the encoder is non-optimal in terms of entropy rate. This provides some leeway to further compress the code with lossless entropy coding. Generally, there are two kind of entropy coding methods includes Huffman tree and arithmetic coding. Between them, arithmetic coding can exhibit better compression rate with a well-defined context, and is adopted in this work.

The binary arithmetic coding is applied according to the CABAC [13] framework. Note that CABAC is originally proposed for video compression. To encode binary code, we modify the coding schedule, redefine the context which leads to the importance mask, and use convolutional neural network for probability prediction. As to coding schedule, we simply code each binary bit map from left to right and row by row, and skip those bits with the corresponding important mask value of zero.

We also extend the convolutional entropy encoder to the quantized importance map. To utilize binary arithmetic coding, a number of binary code maps are adopted to represent the quantized importance map. The convolutional entropy encoder is then trained to compress the binary code maps.

2.5 Model Formulation

In general, the proposed image compression system can be formulated as a rate-distortion optimization problem. Our objective is to minimize the combination of the

distortion loss and rate loss. A tradeoff parameter γ is introduced for balancing compression rate and distortion. Let X be a set of train data, and $x \in X$ be an image from the set. Therefore, the objective function is defined as

$$L = \sum_{x \in X} \{L_D(c, w, A, B, x) + \gamma L_R(x, \mathrm{y})\} \quad \text{s.t. } A \circ F_y^l + F_x^{l-1}(x) \approx w_x^{l-1}(x) + B^{l-1} \quad (9)$$

where c is the code of the input image x and w is the weights between features of input image and features of corresponding side information y. $L_D(c, w, A, B, x)$ denotes the distortion loss and $L_R(x, \mathrm{y})$ denotes the rate loss.

Benefited from the relaxed rated loss and the straight-though estimator of the gradient, the whole compression system can be trained in an end to end manner with ADAM solver. We initialize the model with the parameters pre-trained on training set without the side information. The model is further trained with the learning rate of le^{-4}, le^{-5} and le^{-6}. In each learning rate, the model is trained until the objective function does not decrease. And a smaller learning rate is adopted to fine-tune the model.

3 Experimental Result

Our reference resource-based image compression model is trained on a subset of INRIA Holiday dataset [14] with about 1000 high quality images. We divide these images into 128×128 patches and take use of these patches to train the network. The side information generated from reference resource is the same at encoder and decoder. After training, we test the proposed model on another subset of INRIA Holiday dataset. The compression rate of our model is evaluated by the metric bits per pixel (bpp), which is calculated as the total amount of bits used to code the image divided by the number of pixels. The image distortion is evaluated with Peak Signal to Noise Ratio (PSNR).

3.1 Parameter Setting

In our experiments, we set the number of binary feature maps according to the compression rate. For instance, it will be set as 64 when the compression rate is less than 0.5 bpp and 128 otherwise. Then, the number of importance level is chosen based on importance mask. For $\mathrm{n} = 64$ and $\mathrm{n} = 128$, we set the number of importance level to be 16 and 32, respectively. Moreover, different values of the tradeoff parameter γ in the range [0.0001,0.2] are chosen to get different compression rates. For the choice of the threshold value r, we set it as $r_0 hw$ for $\mathrm{n} = 64$ and $0.5 r_0 hw$ for $\mathrm{n} = 128$. Here, r_0 is the wanted compression rate represent with bit per pixel (bpp).

3.2 Quantitative Evaluation

We compare our model with JPEG, JPEG 2000 and Li et al. [8]. Among different variants of JPEG, the optimized JPEG with 4:2:0 Chroma subsampling is adopted.

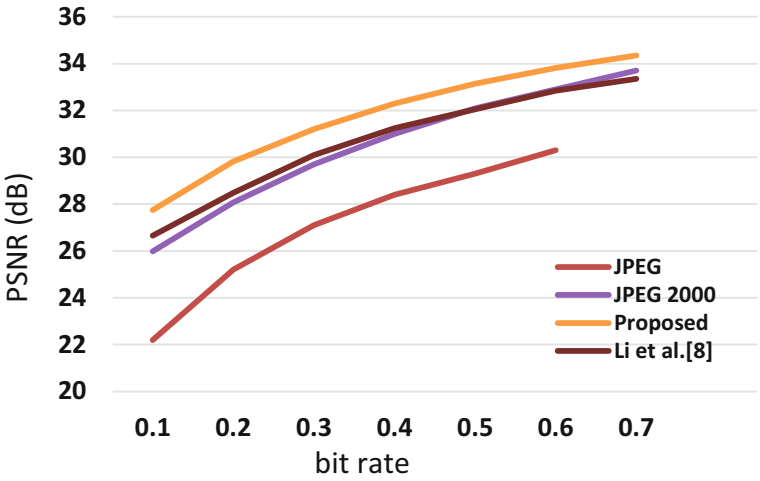

Fig. 5. Comparison of the ratio-distortion curves

(a) JPEG

(b) JPEG 2000

(c) Li et al.[8]

(d) Proposed

Fig. 6. Visual quality evaluation

Using PSNR as performance metric, Fig. 5 gives the ratio-distortion curves of these four methods. In terms of PSNR, the results by Li et al. [8], JPEG 2000 and ours are much higher than that by JPEG. The proposed framework achieves 1.5 dB and 1.2 dB gains in PSNR compared against JPEG 2000 and Li et al. [8].

3.3 Visual Quality Evaluation

In Fig. 6, one can see that the proposed compression framework achieves much better subjective performance than JPEG and JPEG 2000, especially at a very low bit rate. Our framework preserves more high-frequency information and recovers sharp edges and pure textures in the reconstructed image.

4 Conclusion

A convolutional neural network based system is developed for reference resource-based image compression. It well solves the problem of current image compression schemes that is hard to utilize external images for compression even if highly correlated image can be found in the reference resource. With the side information generated from the reference resource, we introduce a neural network architecture, in which the input image and corresponding image are taken as the inputs. With the importance map, we suggest a non-entropy based loss for rate control. Experiments clearly show the superiority of our method in retaining structures and removing artifacts, leading to remarkable visual quality.

References

1. Ghanbari, M.: Standard codecs: Image compression to advanced coding. Iet, no. 49 (2003)
2. Courbariaux, M., Hubara, I., Soudry, D., EI-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1. arXiv:1602.02830 (2016)
3. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_13
4. Girshick, R., Donahue, J., Darrel, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
5. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (2015)
6. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-Net: ImageNet classification using binary convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 525–542. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_32
7. Toderici, G., et al.: Variable rate image compression with recurrent neural networks. arXiv:1511.06085 (2015)

8. Li, M., Zuo, W., Gu, S., Zhao, D., Zhang, D.: Learning convolutional networks for content weighted image compression. arXiv:1703.10553 (2017)
9. Theis, L., Shi, W., Cunningham, A., Husazár, F.: Lossy image compression with compressive autoencoders. arXiv:1703.00395 (2017)
10. Balle, J., Laparra, V., Simoncelli, E.P.: End-to-End optimized image compression. axXiv: 1611.01704 (2016)
11. Chandrasekhar, V., et al.: Tandsform coding of image feature descriptors. In: Proceeding SPIE Conference on Visual Communication and Image Processing, vol. 7257 (2009)
12. Zhou, W.G., Lu, Y.J., Li, H.Q., Song, Y.B., Tian, Q.: Spatial coding for large scale partial-duplicate web image search. In: Proceeding ACM Multimedia, pp. 511–520 (2010)
13. Marpe, D., Schwarz, H., Wiegand, T.: Context-based adaptive binary arithmetic coding in the H.264/avc video compression standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 620–636 (2003)
14. Jegou, H., Douze, M.: INRIA Holiday Dataset (2008). http://lear.inrialpes.fr/people/jegou/data.php