DSSLIC: DEEP SEMANTIC SEGMENTATION-BASED LAYERED IMAGE COMPRESSION

Mohammad Akbari, Jie Liang
akbari@sfu.ca, jiel@sfu.ca*

School of Engineering Science, Simon Fraser University, Canada

Jingning Han
jingning@google.com

Google Inc.

ABSTRACT

Deep learning has revolutionized many computer vision fields in the last few years, including learning-based image compression. In this paper, we propose a deep semantic segmentation-based layered image compression (DSSLIC) framework in which the segmentation map of the input image is obtained and encoded as the base layer of the bit-stream. A compact representation of the input image is also generated and encoded as the first enhancement layer. The segmentation map and the compact version of the image are then employed to obtain a coarse reconstruction of the image. The residual between the input and the coarse reconstruction is additionally encoded as another enhancement layer. Experimental results show that the proposed framework outperforms the H.265/HEVC-based BPG and other codecs in both PSNR and MS-SSIM metrics in RGB domain. Besides, since semantic map is included in the bit-stream, the proposed scheme can facilitate many other tasks such as image search and object-based adaptive image compression[1].

Index Terms— deep learning, semantic segmentation, image compression, generative adversarial networks

1. INTRODUCTION AND PREVIOUS WORKS

Since 2012, deep learning (DL) has revolutionized many computer vision fields such as image classification, object detection, and face recognition. In the last couple of years, it has also made some impacts to the well-studied topic of image compression, and in some cases has achieved better performance than JPEG2000 and the H.265/HEVC-based BPG image codec [1, 2, 3, 4, 5, 6, 7, 8, 9], making it a very promising tool for the next-generation image compression.

Various learning-based image compression frameworks have been proposed. In [1, 2], long short-term memory (LSTM)-based recurrent neural networks (RNNs) were used to extract binary representations, which were then compressed with entropy coding. Johnston et al. [3] utilized structural similarity (SSIM) loss [10] and spatially adaptive bit allocation to further improve the performance. In [4], a scheme that involved a generalized divisive normalization (GDN)-based nonlinear analysis transform, a uniform quantizer, and a nonlinear synthesis transform were developed. Theis et al. [5] proposed a compressive autoencoder (AE) where the quantization was replaced by a smooth approximation, and a scaling approach was used to get different rates. In [6], a soft-to-hard vector quantization approach was introduced, and a unified formulation was developed for image compression.

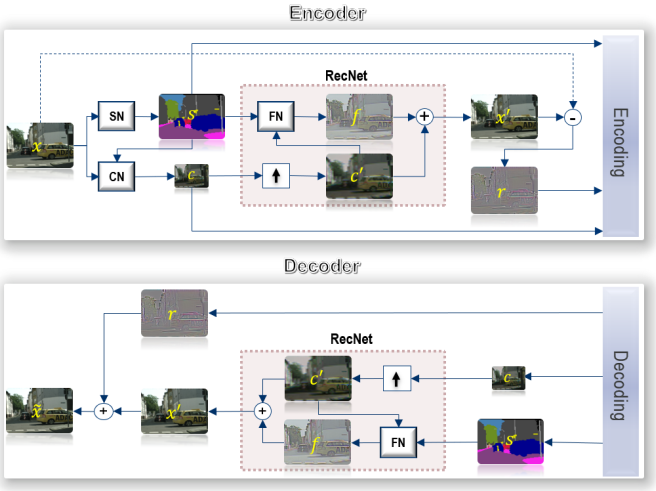

Fig. 1. The overall framework of the proposed deep semantic segmentation-based layered image compression (DSSLIC) codec.

Recently, there have also been some efforts in combining some computer vision tasks and image compression in one framework. In [11, 12], the authors tried to use the feature maps from learning-based image compression to help other tasks such as image classification and semantic segmentation although the results from other tasks were not used to help the compression part. In [9], a segmentation map-based image synthesis model was proposed, which targeted extremely low bit rates (< 0.1 bits/pixel), and used synthesized images for non-important regions.

An advantage of DL is that it can extract much more ac-

*This work is supported by Google Chrome University Research program and the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant RGPIN312262 and RGPAS478109.

[1]The source code of the paper: https://github.com/makbari7/DSSLIC

curate segmentation map from a given image than traditional methods [13]. Recently, it was further shown that DL can even synthesize a high-quality image using only a segmentation map as input [14], thanks to the generative adversarial networks (GAN) [15]. This suggests the possibility of developing efficient image compression using DL-based semantic segmentation and the associated image synthesis.

In this paper, we employ GAN to propose a deep semantic segmentation-based layered image compression (DSSLIC) framework as shown in Figure 1. The idea of semantic segmentation-based compression was already studied in MPEG-4 object-based video coding in the 1990's [16]. However, due to the lack of high-quality and fast segmentation methods, object-based image/video coding has not been widely adopted. Thanks to the rapid development of DL algorithms and hardware, it is now the time to revisit this approach.

In our approach, the segmentation map of the input image is extracted to be losslessly encoded as the base layer of the bit-stream. Next, the input image and the segmentation map are used to obtain a low-dimensional compact representation of the input, which is encoded into the bit-stream as the first and main enhancement layer. The compact image and the segmentation map are then used to obtain a coarse reconstruction of the image. The residual between the input and the coarse reconstruction is encoded as the second enhancement layer. To improve the quality, the synthesized image from the segmentation map is designed to be a residual itself, which aims to compensate the difference between the upsampled compact image and the input image. Therefore, the proposed scheme includes three layers of information.

Experimental results in the RGB (4:4:4) domain show that the proposed framework outperforms the BPG codec [17] in both PSNR and multi-scale structural similarity index (MS-SSIM) [18] metrics across a large range of bit rates, and is much better than JPEG, JPEG2000, and WebP [19]. Moreover, since segmentation map is included in the bit-stream, the proposed scheme can facilitate many other tasks such as image search and object-based adaptive image compression.

2. DEEP SEMANTIC SEGMENTATION-BASED LAYERED IMAGE COMPRESSION (DSSLIC)

The overall framework of the DSSLIC codec is shown in Fig. 1. The encoder includes three DL networks: Segmentation Net, Compact Net, and Fine Net, respectively denoted by SN, CN, and FN. The semantic map s of the input image x is first obtained using SN. In this paper, a pre-trained PSPNet proposed in [13] is used as SN. To help image synthesis, a side information is added to FN, which is obtained from a low-dimensional version c of the original image. In this paper, both s and c are losslessly encoded using the FLIF codec [20], which is a state-of-the-art lossless image codec.

Given the segmentation map s and compact image c, the $RecNet$ part tries to obtain a high-quality reconstruction of the input image. Inside the $RecNet$, c is first upsampled, which, together with s, is fed into FN. FN is trained to learn the missing fine information of the upsampled version of c with respect to the input image. After adding the upsampled version of c and the FN's output f, we get a better estimate of the input. In our scheme, if the SN fails to assign any label to an area, the FN will ignore the semantic input and only reconstruct the image from c, which can still get good results. Therefore, our scheme is applicable to all general images. The residual r between the input and the estimate is then obtained and encoded by a lossy codec. In order to deal with negative values, the residual image r is rescaled to [0, 255] with min-max normalization before encoding. The min and max values are also sent to decoder for inverse scaling. In this paper, the H.265/HEVC intra coding-based BPG codec is used [17], which is state-of-the-art in lossy coding. As a result, in our scheme, the segmentation map s serves as the base layer, and the compact image c and the residual r are respectively the first and second enhancement layers.

At the decoder side, the segmentation map and compact representation are decoded to be used by $RecNet$ to get an estimate of the input image. The output of $RecNet$ is then added to the decoded residual image to get the final reconstruction $\tilde{x}$.

The architectures of the CN (proposed in this work) and FN (modified from [14]) networks are respectively defined as $\{c_{64}, c_{128}, c_{256}, c_{512}, c_3, tanh\}$ and $\{c_{64}, c_{128}, c_{256}, c_{512}, 9\times r_{512}, u_{256}, u_{128}, u_{64}, c_3, tanh\}$, where $\boldsymbol{c_k}$ denotes a 3×3 convolution layer (with k filters and stride one) followed by instance normalization and ReLU. The filter size for the first and last layers is 7×7; $\boldsymbol{r_k}$ is a residual block containing reflection padding and two 3×3 convolution layers (with k filters) followed by instance normalization; and $\boldsymbol{u_k}$ is a 3×3 fractional-strided-convolution layer (with k filters and stride $\frac{1}{2}$) followed by instance normalization and ReLU.

Inspired by [14], for the adversarial training of the proposed model, two discriminators denoted by D_1 and D_2 operating at two different image scales are used in this work. D_1 operates at the original scale and has a more global view of the image. Thus, the generator can be guided to synthesize fine details in the image. On the other hand, D_2 operates with 2× down-sampled images, leading to coarse information in the synthesized image. Both discriminators have the following architecture: $\{C_{64}, C_{128}, C_{256}, C_{512}\}$, where C_k denotes 4×4 convolution layers with k filters and stride 2 followed by instance normalization and LeakyReLU. In order to produce a 1-D output, a convolution layer with one filter is utilized after the last layer of the discriminator.

2.1. Formulation and Objective Functions

Let $x \in R^{h\times w\times k}$ be the original image, the corresponding segmentation map $s \in Z^{h\times w}$ and the compact representation

$c \in R^{\frac{h}{\alpha} \times \frac{w}{\alpha} \times k}$ are generated as follows: $s = SN(x), c = CN(s, x)$.

Conditioned on s and the upscaled c, denoted by $c' \in R^{h \times w \times k}$, FN (our GAN generator) reconstructs the fine information image, denoted by $f \in R^{h \times w \times k}$, which is then added to c' to get the estimate of the input: $x' = c' + f$, where $f = FN(s, c')$.

The error between x and x' is measured using a combination of different losses including $\mathcal{L}_1$, $\mathcal{L}_{SSIM}$, $\mathcal{L}_{DIS}$, $\mathcal{L}_{VGG}$, and GAN losses. The L1-norm loss (least absolute errors) is defined as: $\mathcal{L}_1 = 2\lambda \|x - x'\|_1$. It has been shown that combining pixel-wise losses such as $\mathcal{L}_1$ with SSIM loss can significantly improve the perceptual quality of the reconstructed images [21]. As a result, we also utilize the SSIM loss denoted by $\mathcal{L}_{SSIM}$ in our work.

To stabilize the training of the generator and produce natural statistics, two perceptual feature-matching losses based on the discriminator and VGG networks [22] are employed. The discriminator-based loss is calculated as:

$$\mathcal{L}_{DIS} = \lambda \sum_{d=1,2} \sum_{i=1}^{n} \frac{1}{N_i} \|D_d^{(i)}(s, c', x) - D_d^{(i)}(s, c', x')\|_1, \tag{1}$$

where $D_d^{(i)}$ denotes the features extracted from the i-th intermediate layer of the discriminator network D_d (with n layers and N_i number of elements in each layer). Similar to [7], a pre-trained VGG network with m layers and M_j elements in each layer is used to construct the VGG perceptual loss as in below:

$$\mathcal{L}_{VGG} = \lambda \sum_{j=1}^{m} \frac{1}{M_j} \|V^{(j)}(x) - V^{(j)}(x')\|_1, \tag{2}$$

where V_j represents the features extracted from the j-th layer of VGG. In order to distinguish the real training image x from the reconstructed image x', given s and c', the following objective function is minimized by the discriminator D_d:

$$\mathcal{L}_D = -\sum_{d=1,2} (\log D_d(s, c', x) + \log(1 - D_d(s, c', x'))), \tag{3}$$

while the generator (FN in this work) tries to fool D_d by minimizing $-\sum_{d=1,2} \log D_d(s, c', x')$. The final generator loss including all the reconstruction and perceptual losses is then defined as:

$$\mathcal{L}_G = -\sum_{d=1,2} \log D_d(s, c', x') + \mathcal{L}_1 + \mathcal{L}_{SSIM} + \mathcal{L}_{DIS} + \mathcal{L}_{VGG}. \tag{4}$$

Finally, our goal is to minimize the hybrid loss function $\mathcal{L} = \mathcal{L}_D + \mathcal{L}_G$.

3. EXPERIMENTS

The ADE20K dataset with 150 semantic labels [23] is used for training the proposed model. The images with at least 512 pixels in height or width are used (9272 images in total). All images are rescaled to $h = 256$ and $w = 256$ to have a fixed size for training. Note that no resizing is needed for the test images since the model can work with any size at the testing time. We set the downsampling factor $\alpha = 8$ to get the compact representation of size $32 \times 32 \times 3$. We also consider the weight $\lambda = 10$ for $\mathcal{L}_1$, $\mathcal{L}_{DIS}$, and $\mathcal{L}_{VGG}$.

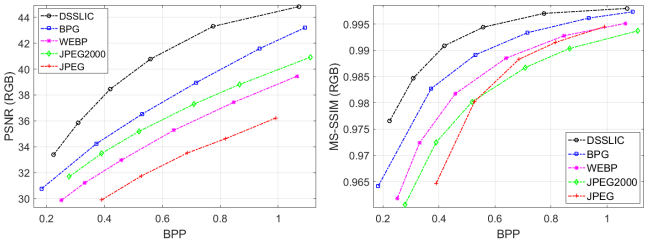

Fig. 2. Comparison results on ADE20K test set.

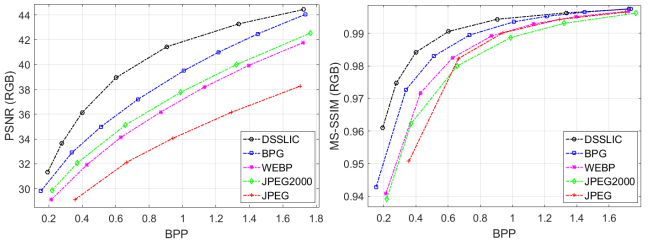

Fig. 3. Comparison results on Kodak image set.

All models were jointly trained for 150 epochs with mini-batch stochastic gradient descent (SGD) and a mini-batch size of 8. The Adam solver with learning rate of 0.0002 was used, which is fixed for the first 100 epochs, but gradually decreases to zero for the next 50 epochs. Perceptual feature-matching losses usually guide the generator towards more synthesized textures in the predicted images, which causes a slightly higher pixel-wise reconstruction error, especially in the last epochs. To handle this issue, we did not consider the perceptual $\mathcal{L}_D$ and $\mathcal{L}_{VGG}$ losses in the generator loss for the last 50 epochs. All the SN, CN, FN, and the discriminator networks proposed in this work are trained in the RGB domain.

We compare the performance of the proposed DSSLIC scheme with JPEG, JPEG2000, WebP, and the BPG codec [17], which is state-of-the-art in lossy image compression. Since the networks are trained for RGB images, we encode all images using RGB (4:4:4) format in different codecs for fair comparison. We use both PSNR and MS-SSIM [18] as the evaluation metric in this experiment. In this experiment, we encode the RGB components of the residual image r using lossy BPG codec with different quantization values.

The results of the ADE20K test set (averaged over 50 random test images not included in the training set) are given in Figure 2. To demonstrate the generalization capability of

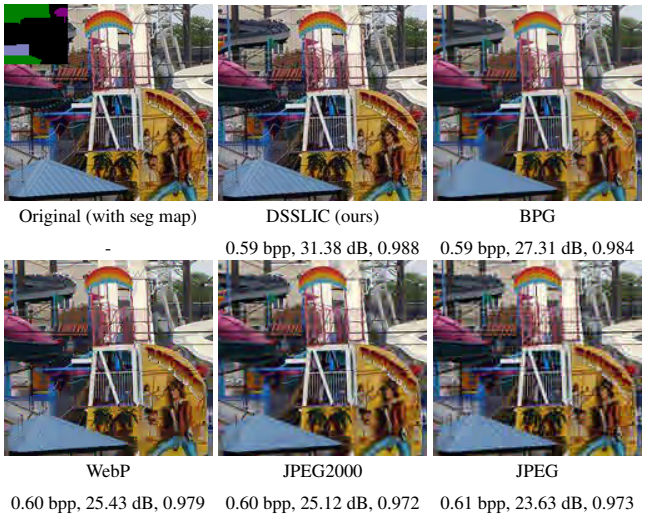

Fig. 4. ADE20K visual example (BPP, PSNR, MS-SSIM)

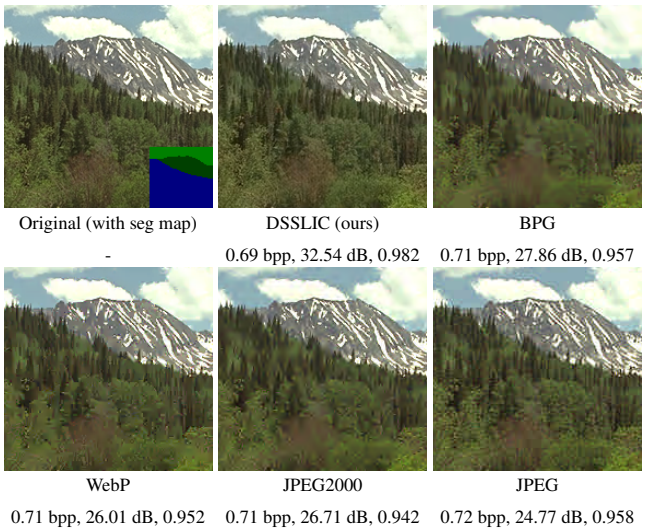

Fig. 5. Kodak visual example (BPP, PSNR, MS-SSIM)

the scheme, the ADE20K-trained model is also applied to the classical Kodak dataset (including 24 test images) and reported in Figure 3. As shown in the figures, our method gives better PSNR and MS-SSIM than other codecs. In particular, the PSNR gains over BPG can be 2-4 dB in the middle bitrate ranges. Some visual examples from ADE20K and Kodak test sets are given in Figures 4 and 5.

Figure 6 and Table 1 report some ablation studies of different configurations, all are obtained without using the BPG-based residual coding, including: **upComp**: the results are obtained without considering the FN network in the pipeline, i.e., $x' = c'$ (the upsampled compact image only); **noSeg**: the segmentation maps are not considered in neither CN nor FN networks, i.e., $x' = c' + f$ where c' is the upsampled version of $c = CN(x)$, and $f = FN(c')$; **withSeg**: all the DSSLIC

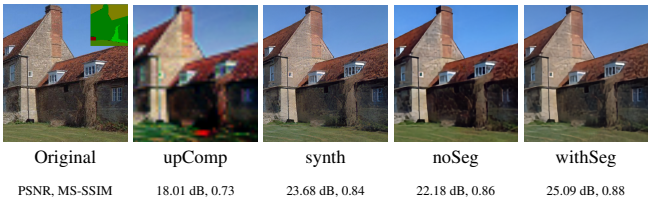

Fig. 6. Visual comparison of different scenarios at 0.08 BPP.

Table 1. Results of different scenarios (without BPG-based residual coding).

	ADE20K				Kodak			
	upComp	**synth**	**noSeg**	**withSeg**	**upComp**	**synth**	**noSeg**	**withSeg**
BPP	0.095	0.092	0.08	0.095	0.087	0.088	0.080	0.087
PSNR	17.50	21.91	22.24	23.11	17.77	20.97	21.46	21.91
MS-SSIM	0.759	0.887	0.905	0.914	0.738	0.858	0.887	0.891

components shown in Figure 1 are used in this configuration (except BPG-based residual coding); **synth**:, the settings in this configuration is the same as withSeg except that the perceptual losses $\mathcal{L}_{VGG}$ and $\mathcal{L}_{DIS}$ are considered in all training epochs. The poor performance of using only the upsampled compact images in **upComp** shows the importance of FN in predicting the missing fine information, which is also visually obvious in Figure 6. Considering perceptual losses in all training epochs (**synth**) leads to sharper and perceptually more natural images, but the PSNR is much lower. The results with segmentation maps (**withSeg**) provide slightly better PSNR than **noSeg** although the visual gain is more pronounced, e.g., the dark wall in Figure 6.

In overall, our approach preserves more details and provides results with higher visual quality compared to BPG and other codecs, which demonstrates the great potential of DL-based image compression. In addition, the built-in semantic map enables some new applications such as fast content-based image retrieval, object-based video coding, and region-of-interest coding.

4. CONCLUSION

In this paper, we proposed a deep semantic segmentation-based layered image compression (DSSLIC) framework in which the semantic map of the input image was used to synthesize the image, and a compact representation and a residual were encoded as enhancement layers in the bit-stream. Experimental results showed that the proposed framework outperforms the H.265/HEVC-based BPG and the other standard codecs in both PSNR and MS-SSIM metrics in RGB (4:4:4) domain. In addition, since semantic map is included in the bit-stream, the proposed scheme can facilitate many other tasks such as image search and object-based adaptive image compression. The proposed scheme opens up many future topics, for example, modifying the scheme for YUV-coded images and applying the framework for other tasks.

5. REFERENCES

[1] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.

[2] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 5435–5443.

[3] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, Sung J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," *arXiv preprint arXiv:1703.10114*, 2017.

[4] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[5] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

[6] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *arXiv preprint arXiv:1704.00648*, 2017.

[7] S. Santurkar, D. Budden, and N. Shavit, "Generative compression," *arXiv preprint arXiv:1703.01467*, 2017.

[8] O. Rippel and L. Bourdev, "Real-time adaptive image compression," *arXiv preprint arXiv:1705.05823*, 2017.

[9] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," *arXiv preprint arXiv:1804.02958*, Apr. 2018.

[10] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[11] S. Luo, Y. Yang, and M. Song, "DeepSIC: Deep semantic image compression," *arXiv preprint arXiv:1801.09468*, 2018.

[12] R. Torfason, F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Towards image understanding from deep compression without decoding," *arXiv preprint arXiv:1803.06131*, 2018.

[13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[14] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," *arXiv preprint arXiv:1711.11585*, 2017.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[16] R. Talluri, K. Oehler, T. Bannon, J. Courtney, A. Das, and J. Liao, "A robust, scalable, object-based video compression technique for very low bit-rate coding," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 7, no. 1, pp. 221–233, 1997.

[17] F. Bellard, "Bpg image format (http://bellard.org/bpg/)," 2017.

[18] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*. Ieee, 2003, vol. 2, pp. 1398–1402.

[19] Google Inc., "WebP (https://developers.google.com/speed/webp/)," 2016.

[20] J. Sneyers and P. Wuille, "FLIF: Free lossless image format based on maniac compression," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 66–70.

[21] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, vol. 1, p. 4.