A new end-to-end image compression system based on convolutional neural networks

Pinar Akyazi, Touradj Ebrahimi

Pinar Akyazi, Touradj Ebrahimi, "A new end-to-end image compression system based on convolutional neural networks," Proc. SPIE 11137, Applications of Digital Image Processing XLII, 111370M (6 September 2019); doi: 10.1117/12.2530195

SPIE.

Event: SPIE Optical Engineering + Applications, 2019, San Diego, California, United States

A new end-to-end image compression system based on convolutional neural networks

Pinar Akyazi and Touradj Ebrahimi

Multimedia Signal Processing Group (MMSPG)
Ecole Polytechnique Fédérale de Lausanne
CH 1015, Lausanne, Switzerland

ABSTRACT

In this paper, two new end-to-end image compression architectures based on convolutional neural networks are presented. The proposed networks employ 2D wavelet decomposition as a preprocessing step before training and extract features for compression from wavelet coefficients. Training is performed end-to-end and multiple models operating at different rate points are generated by using a regularizer in the loss function. Results show that the proposed methods outperform JPEG compression, reduce blocking and blurring artifacts, and preserve more details in the images especially at low bitrates.

Keywords: Learning-based image compression, low-rate image compression, deep convolutional neural networks, discrete wavelet transform.

1. INTRODUCTION

Image compression is a challenging problem that has been drawing the attention of both researchers and multimedia service providers. The main goal of image compression is to deliver an as high visual quality as possible while maintaining the bitrates reasonably low, depending on the system requirements. Image and video processing communities have been proposing different solutions to improve compression efficiency. Traditional image compression algorithms use hand-crafted features and fixed transforms to represent the encoded bitstreams. Recently, the performance of learning-based image compression models have started to reach that of state-of-the-art transform-based approaches.

The most widely used traditional codecs include JPEG,[1] JPEG 2000,[2] WebP[3] and HEVC/H.265 intra.[4] Developed by the Joint Photographic Experts Group (JPEG), JPEG is a Huffman-coded discrete cosine transform (DCT) based lossy image compression approach, which was developed more than two decades ago. JPEG 2000 was created with the intention of succeeding JPEG, using wavelet transform instead of DCT. The blur artifacts of wavelet transform were more pleasant to human visual system when compared to the blocking artifacts of the DCT. The most recent standard being developed by the JPEG community, JPEG XL,[5] includes many enhancements to previous versions, such as variable-size DCT, nonlinear Haar transforms, multiresolution encoding, adaptive quantization, adaptive loop filters and context modeling. HEVC/H.265 intra is a more recent standard that employs variable block size segmentation and intra prediction within the same picture. Such complex improvements, however, increase the encoding time drastically. WebP also has improving features over JPEG standards such as prediction coding and block adaptive quantization, while remaining computationally efficient at the encoding and decoding.

Machine learning approaches have demonstrated advanced solutions to many image processing problems such as object classification and image enhancement, and are continually improving at image related tasks. Recent learning-based image compression models have reached and even surpassed the performance of transform-based state-of-the-art image codecs. Learning-based methods do not employ hand-crafted features but rather extract a latent representation of the input image through the use of training neural networks. Recent learning-based image compression methods involve different neural network architectures, such as convolutional neural networks (CNN), recurrent neural networks (RNN) and generative adversarial networks (GAN). Convolutional neural

Further author information: (Send correspondence to authors) E-mail: firstname.lastname@epfl.ch

Applications of Digital Image Processing XLII, edited by Andrew G. Tescher, Touradj Ebrahimi, Proc. of SPIE
Vol. 11137, 111370M · © 2019 SPIE · CCC code: 0277-786X/19/$21 · doi: 10.1117/12.2530195

networks extract local features of the image at each layer while recurrent neural networks allow processing the images in a sequential manner. Generative models, on the other hand, create new data from training images by learning the statistics. All these models can be efficient for the task of image compression by training a network end-to-end, provided that the loss function measures a meaningful distortion between the original and the decoded images. Different models will be discussed in detailed in the next section.

In this paper, two end-to-end autoencoders (ResWCAE and ResMixWCAE) using convolutional neural networks are proposed. The main novelty of this work lies in the input of the networks, where instead of images themselves their Haar wavelet coefficients are used after a three level decomposition. To the authors' best knowledge, such preprocessing method has not been employed except for,[6] where it was shown that wavelet decomposition help the network achieve higher qualities when compared to no preprocessing. The methods presented are extensions to WCAE presented in[6] with modifications to the architecture and increased range of rate points. The latent representations in all models are constructed by the convolutional neural networks and then quantized and entropy coded. The models trained end-to-end using approximations for the discrete quantization step and entropy estimation. The loss function measures the mean squared error between the original and decoded images, plus a rate term. The performance of the proposed methods are evaluated on the test set of CVPR Workshop and Challenge on Learned Image Compression 2019 (CLIC2019),[7] and compared to the performance of JPEG, JPEG 2000 and WebP. Results show that the proposed models outperform JPEG and manage to preserve the high frequency details in the images after compression. Compared to JPEG 2000 and WebP, however, the high frequency noise decreases subjective quality at higher bitrates.

The rest of this paper is organized as follows: The following section describes state-of-the-art approaches to learning-based image compression. Proposed framework is presented in detail in Section 3, followed by experiments and results in Section 4. Conclusions are delivered in Section 5, along with possible improvements and future directions.

2. RELATED WORK

Learning-based methods have been involved in image compression in components such as learning more efficient frequency transforms, predictive coding, segmentation and quantization.[8] More recent methods tackle the entire compression problem in end-to-end models using autoencoder architectures. These autoencoders have three main parts: (i) the encoder where the input is mapped to the latent space, (ii) the bottleneck where the latent representation is coded and (iii) the decoder, where the code is transformed back to yield an output close to the input. A typical bottleneck involves quantization of the code followed by an efficient representation such as entropy or arithmetic coding. In convolutional architectures, the decoder usually inverts the encoding process by replacing the convolution filters with reversed operators, i.e. deconvolution. Pooling and activation functions are also inverted during decoding.

A general framework for variable-rate image compression based on convolutional and deconvolutional Long Short-Term Memory (LSTM) recurrent networks is presented in.[9] The proposed network achieves progressive encoding as it is able to deliver more accurate representations by sending more bits. Several architectures have been tested: feed forward fully-connected residual encoder, LSTM-based residual encoder, feed-forward convolutional/deconvolutional residual encoder and convolutional/deconvolutional LSTM compression. The fully connected residual encoder employs a stack of fully connected layers at the encoder and decoder. The residual at stage t is fed into the next stage $t+1$ for progressive encoding using more bits. The stages are also implemented using LSTM blocks, and convolutional/deconvolutional filters. A combination of LSTM blocks and convolutional/deconvolutional filters is also tested. The LSTM compressor and the combined compressor were shown to outperform JPEG on thumbnail size images, in terms of SSIM.

The resolution of images has been increased in[10] and,[11] as well as an increase in the speed of the algorithm. The former work employs convolutional autoencoders (CAEs) where deconvolutions are replaced by sub-pixel convolutions, and residual connections and leaky rectifications between layers of encoder and decoder are added. The latter compares different RNN types (LSTM and associative LSTM) and introduces a hybrid GRU[12] and ResNet[13] model. The extensive comparison between architectures show that it is not easy to pick a winning algorithm, since results vary with respect to different quality metrics and at various bitrates. The authors also

show that training the models on "hard to compress" data yields better quality in terms of MS-SSIM[14] and PSNR-HVS metrics. A trade-off parameter for training the models at different bitrates is introduced in[15] and widely used in other works such as.[16–18] The generalized divisive normalization (GDN)[19] nonlinearity is also shown to increase the performance of the end-to-end image compression model.

Further improvements are implemented in[17] where entropy coding involves an autoencoder that learns a hyperprior, capturing the spatial dependencies within the latent representation. Models using MSE and MS-SSIM loss are evaluated, where experimental results on the Kodak dataset[20] show that the model outperforms BPG, an encapsulation of HEVC/H.265 intra, in terms of MS-SSIM. Other works employ a variety of entropy models including factorized entropy models[6,16,18] and single-iteration and progressive LSTM-based entropy models[11] during training, and range coder[10] and context-based adaptive binary arithmetic framework[21] during test.

At the bottleneck, quantization of the latent representation can also be modeled differently. A widely preferred approach is performing uniform scalar quantization, i.e. rounding to nearest integer, which effectively implements a parametric form of vector quantization on the original image space.[15] This operation is not differentiable and therefore a smooth approximation is implemented by adding uniform noise to the latent representation during training. In,[10] a stochastic rounding operation is defined where the derivative is replaced with the derivative of the expectation in the backward pass. Similarly, a stochastic form of binarization is used in.[9] Another soft relaxation is presented in[22] where vector quantization is explored in the context of learned image compression and improvements over scalar quantization are demonstrated.

At very low bitrates, i.e. bitrates below 0.15bpp, the generative adversarial model presented in[23] achieved good subjective image quality by fully synthesizing selected regions of images. At similar bitrates[18] has shown that residual architectures and deeper networks increase the performance both subjectively and objectively. This paper builds on[6] by adding more filters, residual architecture and increasing the output size of the encoder. The network is trained at various rate points to allow full comparison along a rate-distortion curve with selected state-of-the-art traditional and learning-based compression models.

3. PROPOSED FRAMEWORK

The structure of the proposed architecture is depicted in Figure 1. The input color image X is first separated into non-overlapping patches of dimensions $N \times M$. Before the analysis stage of the convolutional autoencoder, each color channel of an RGB input image patch is first normalized to have $[-1, 1]$ range and then undergoes a 3-scale 2D wavelet transform, where Daubechies-1 wavelets are used. 2D wavelet decomposition is known to be effective in various image processing tasks, compression in particular. When compressing an image using convolutional neural networks, although image features are expected to be learned by the network without ideally any preprocessing, it was shown that such preprocessing step increases the quality of the output.[6] Using wavelet decomposition separates the input image into its high frequency and low frequency components at different scales, allowing more control over the visual characteristics of the compressed image by giving more or less emphasis on particular frequency components.

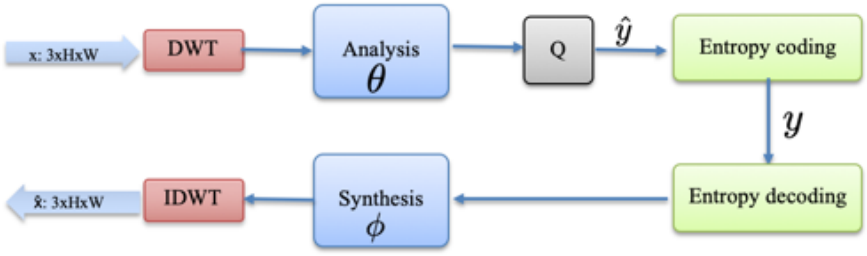


Figure 1: Proposed convolutional autoencoder architecture.

Two different analysis/synthesis blocks have been implemented by modifying WCAE (Figure 2, as depicted in Figure 3. All analysis blocks are separated into three channels for each scale of the wavelet transform. The coarsest scale has 12 inputs, 4 from each of the 3 color components. The second and finest scales have 9 inputs

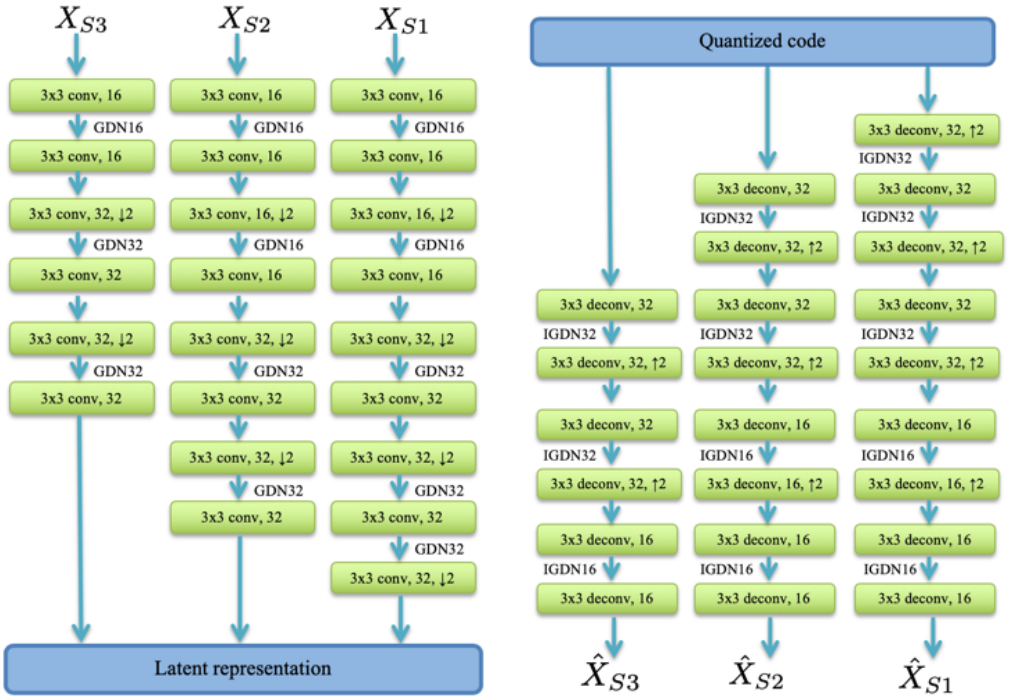


Figure 2: Analysis and synthesis blocks of WCAE.

each. Convolutional filters of dimensions 3×3 are used at each layer of WCAE and the number of outputs is doubled once for all scales. The coarsest, second and finest scales inputs are downsampled 2, 3 and 4 times, respectively. Between the convolutional layers, GDN functions provide a nonlinear mapping of the layer outputs. The latent representation is formed by concatenating the 32 outputs of each scale, and has dimensions $32 \times \frac{N \times M}{1024}$. With this representation, the $3 \times N \times M$ input is reduced approximately by a factor of 33, in size.

The architecture with analysis/synthesis block in Figure 3 is referred to as ResWCAE. ResWCAE has an increased number of outputs compared to WCAE, i.e. 64 outputs at each scale instead of 32, is deeper and has a residual architecture with the use of skip connections.[13] The same architecture is used to create the third model, referred to as ResMixWCAE. All 3×3 convolutional kernels in the middle and finest scales of ResMixWCAE are replaced by 5×5 and 7×7 kernels, respectively.[24] ResWCAE and ResMixWCAE reduce the input approximately by a factor of 15, in size.

Once the latent representation (code) is constructed, it needs to be quantized for all architectures. Since quantization is a function with zero gradients almost everywhere, it is replaced by additive uniform noise during training. This is a method preferred at the quantization step of learning-based encoders[15,16] assuming unit bin size and uncorrelated quantization error between elements. The quantized values then need to be encoded, where the resulting rate will be a component of the overall loss function. Since the latent representation has been uniformly quantized, an effective entropy coding is expected to reduce the rate optimally. However, the entropy coding also needs to be fully differentiable. The lower bound of the rate, on the other hand, is equal to the entropy of the quantized code.[25] It is therefore sufficient to compute the entropy of the quantized code and use it in the loss function as an estimate of the rate portion. Once the entropy is computed, the quantized code then goes through a synthesis stage, where deconvolutional filters and upsampling operators are used in parallel with the analysis stage. The quantized code is separated into three equally dimensional components, which represent the coarsest, second and finest scales of the decoded image at the output of the synthesis transform. The three outputs are then merged using an inverse wavelet transform and yield the decoded image. The distortion between the original and decoded image is used in the loss function. The overall loss function of the of all three architectures is then:

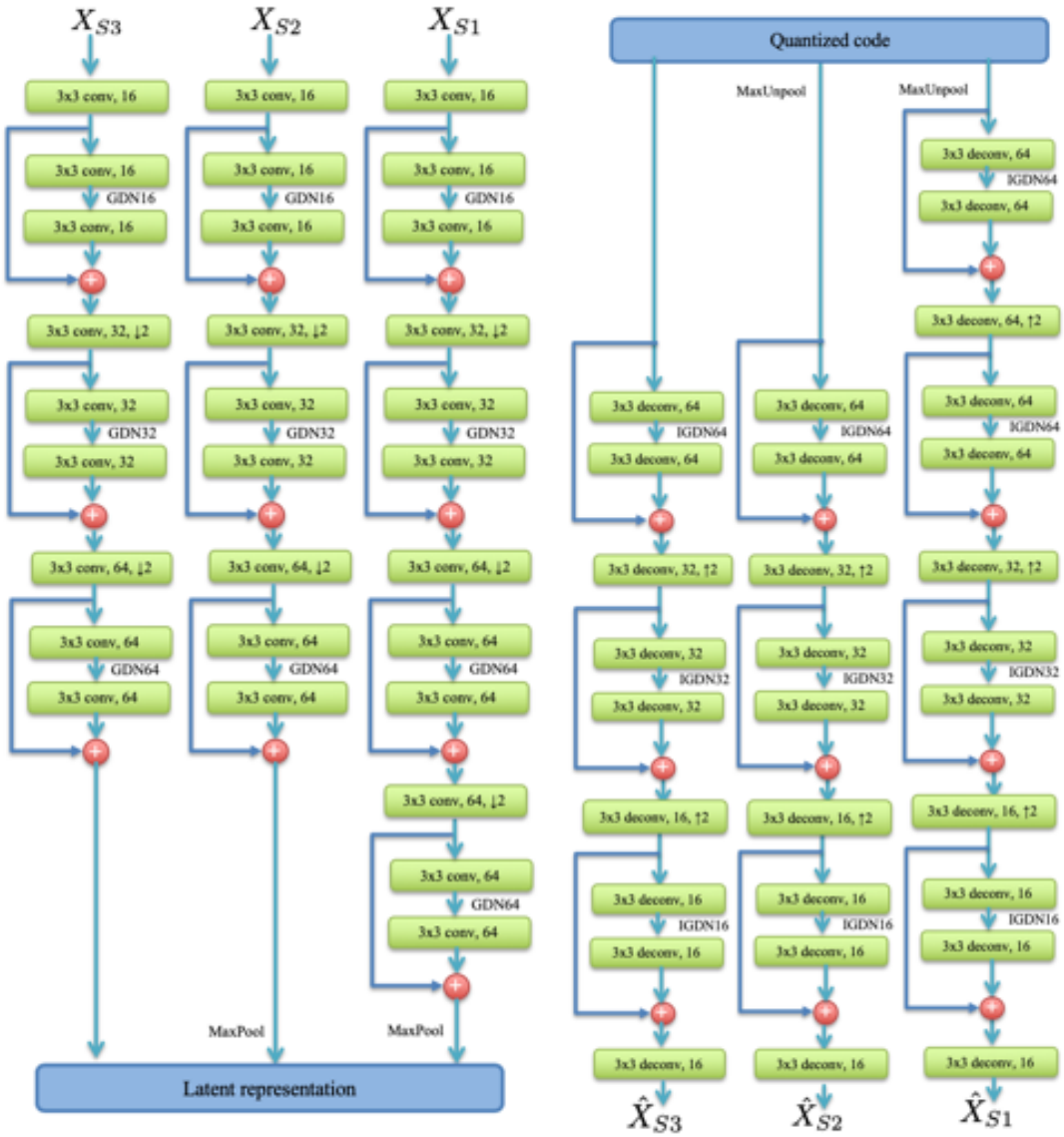


Figure 3: Analysis and synthesis blocks of ResWCAE. The same architecture is used for ResMixWCAE, with the kernel sizes of the middle and fine scales increased to 5 and 7, respectively.

$$J(\theta, \phi; X) = D(X, \hat{X}) + \lambda R \tag{1}$$

where

$$D(X, \hat{X}) = \sum (X - \hat{X})^2 \tag{2}$$

$$R = \sum_i P_{\hat{y}}(\hat{y}_i) \log_2 P_{\hat{y}}(\hat{y}_i) \tag{3}$$

where X is the input image, $\hat{X}$ is the decoded image, D is the distortion, R is the entropy of the quantized code $\hat{y}$ which has the distribution $P_{\hat{y}}(\hat{y})$. Here, the distribution of $\hat{y}$ is approximated to be Gaussian after the use of multiple GDN nonlinearities and the discrete probability function of $\hat{y}$ is computed as a Gaussian distribution with mean $\mu_{\hat{y}}$ and standard deviation $\sigma_{\hat{y}}$. Regularization parameter λ controls the rate, where a larger λ forces the latent representation to have smaller entropy.

After the network is fully trained, the latent representation is quantized by rounding to the nearest integer at test time and entropy encoding is performed by the range encoder.[10] The range encoder expects a positive input, therefore the minimum value of the quantized code is subtracted and then passed to the decoder. In addition, the decoder also needs to receive the input image dimensions, as each input channel is padded with zeros to have dimensions that are multiples of 32. Finally, the cumulative distribution function of the quantized code is also passed to the decoder. The total size of the encoded bitstream is then equal to the sum of these additional parameters sent to the decoder and the output of the range encoder.

4. EXPERIMENTS AND RESULTS

For training the networks, the mobile and professional training datasets of CLIC2019 were used and each image was divided into 256×256 non-overlapping patches. A total of 16750 distinct patches were used during training. The networks were trained iteratively using back propagation[26,27] and the Adam[28] optimizer with a batch size of 8 and learning rate of 10^{-4} were used. Training continued for a total of 100 epochs for each network, where losses had converged to stable values. The parameter λ was tuned to yield target bitrates of [0.12, 0.25, 0.50, 0.75, 1.00] bpp. Early stopping criterion was applied, i.e. models with the lowest validation error were selected as final models for testing.

The results of the proposed methods were compared to three different transform-based codecs, JPEG, JPEG 2000 and WebP, using the objective metrics PSNR,,[29] MS-SSIM,[14] VIF[30] and VMAF.[31] Performance plots for each metric are depicted in Figure 4. All results have been averaged on the complete test dataset, which contained 330 different images.

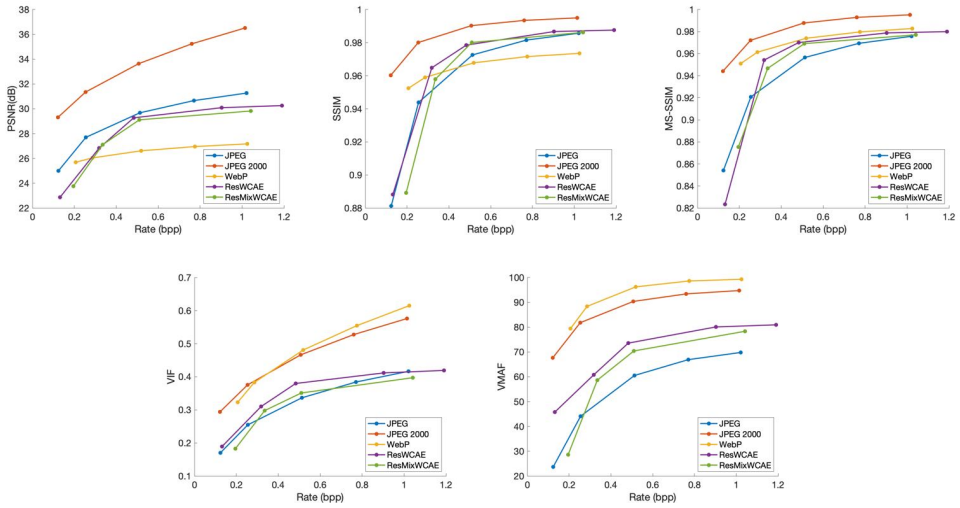
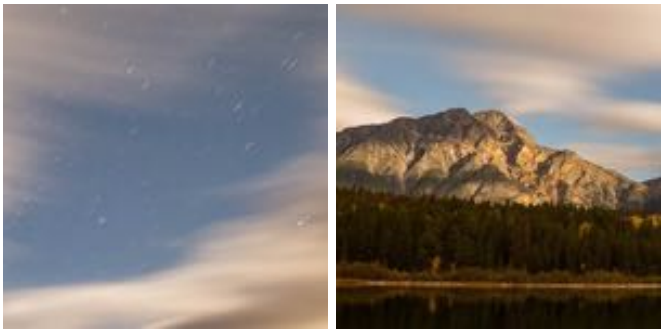


Figure 4: Performances of codecs with respect to selected objective metrics measured at target bitrates.

WCAE was designed for the low bitrate track of the CLIC2019 challenge, and therefore was optimized for bitrates as low as 0.15bpp. The lower number of outputs of the analysis stage of WCAE limit the rate at below 1.00bpp on the test set, when no regularization is applied on the loss function. The latent representation of ResWCAE and ResMixWCAE are almost twice the size of WCAE and consequently perform better at higher bitrates.

Despite the fact that the networks were trained using MSE loss, the PSNR of ResWCAE and ResMixWCAE are lower than JPEG and JPEG 2000 at all target bitrates. The proposed models have higher PSNR than WebP, however this is because images were converted to YCbCr4:2:0 format before encoding with WebP and the conversions add a shift to the decoded WebP image. The visual quality of WebP is superior to the proposed



(a) (b)

(c) (d)

Figure 5: 300×300 cropped regions from test images (a)-(c) and a full test image of resolution 1875×1500.

models, as can be verified in Figures 6-9. On the other hand, the performance of proposed models surpass JPEG in terms of SSIM, MS-SSIM, VIF and VMAF, with ResMixWCAE performing slightly worse than ResWCAE.

Selected examples from the test set are depicted in Figure 5, where 5(a)-(c) have been cropped from two test images and (d) is a whole test image. The corresponding decoded images are presented in Figures 6-9. A closer examination verifies that the proposed methods perform better than JPEG at all target bitrates. JPEG has significant blocking artifacts at the lower bitrates, whereas at the higher bitrates some details are smoothed out as can be seen in 6. A similar effect is observed at the lowest bitrates for WebP, where the contents have been smoothened. This can be seen clearly on Figures 6-8, also with some blocking artifacts on Figure 6. JPEG 2000, on the other hand, suffers from ringing artifacts at the lowest bitrates.

The artifacts of ResWCAE and ResMixWCAE are very different from each other and the transform-based codecs at the lowest bitrate. ResWCAE preserves the high frequency components of the images by sharpening the image, at the cost of adding high frequency noise. ResMixWCAE, has less high frequency noise compared to ResWCAE at the expense of loss in details. Distortions in the color channels can be traced on both ResWCAE and ResMixWCAE images, especially on Figures 6 and 9. Such effects cease as bitrates increase, however some high frequency artifacts remain, especially for ResMixWCAE. This indicates that larger kernels on the finer wavelet scales contribute more to high frequency noise. The number of parameters or WCAE, ResWCAE and ResMixWCAE are 264926, 961886 and 3451998, respectively. The complexities of ResWCAE and ResMixWCAE are approximately 3.5 and 13 times that of WCAE. The performance gap between ResWCAE and ResMixWCAE suggests that the latter needs to be trained on a larger dataset. A more intuitive improvement, however, would be to reverse the order of kernels and investigate the effect of larger kernel sizes on the coarsest scale, as well as all three scales. Another important insight can be obtained by comparing these models to networks with similar architectures, which take images without preprocessing as inputs rather than using wavelet coefficients.

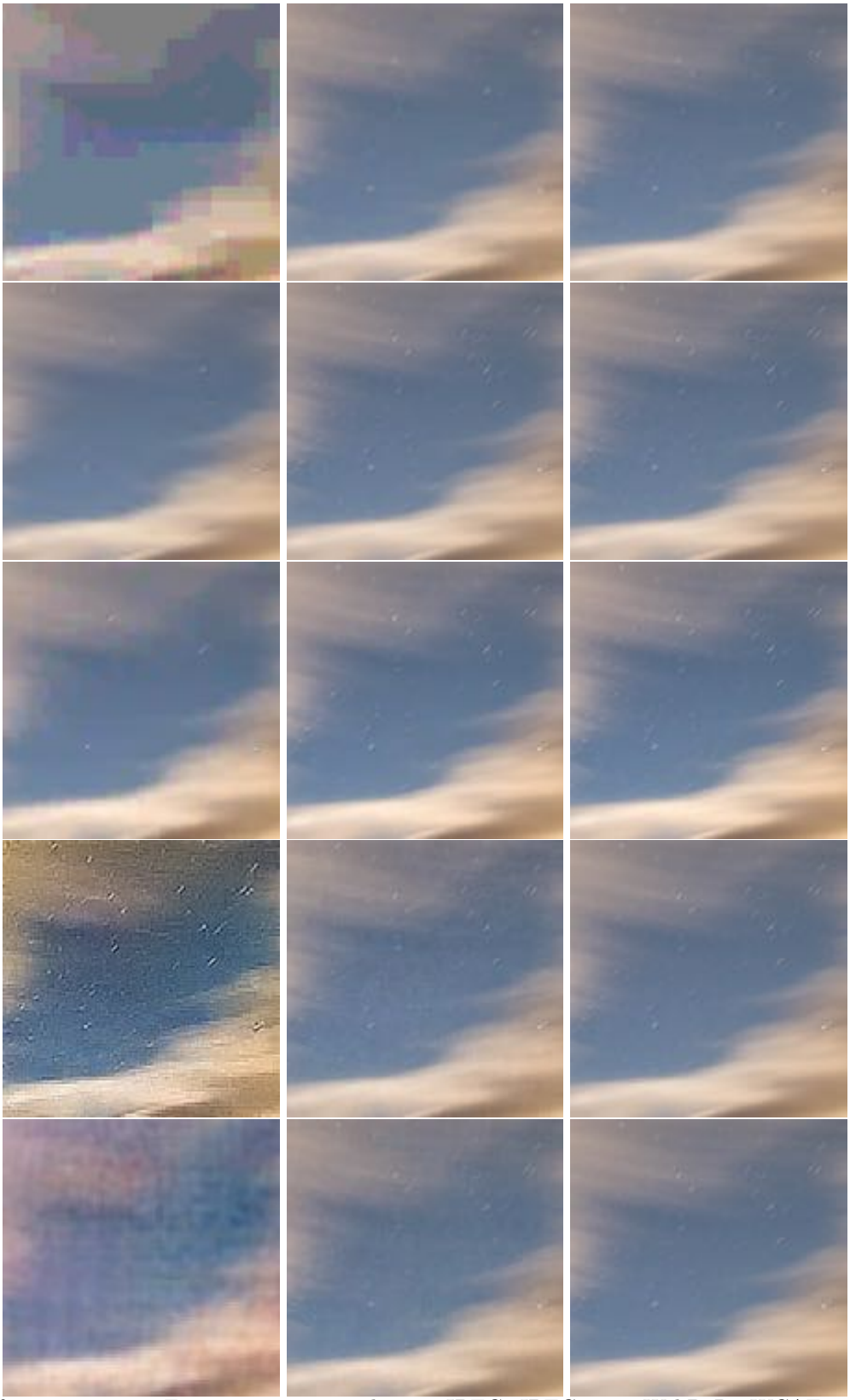


Figure 6: Reference image in Figure 5a compressed using JPEG, JPEG 2000, WebP, ResWCAE and ResMixW-CAE from top to bottom, at target bitrates 0.12, 0.50 and 1.00 bpp from left to right, respectively.

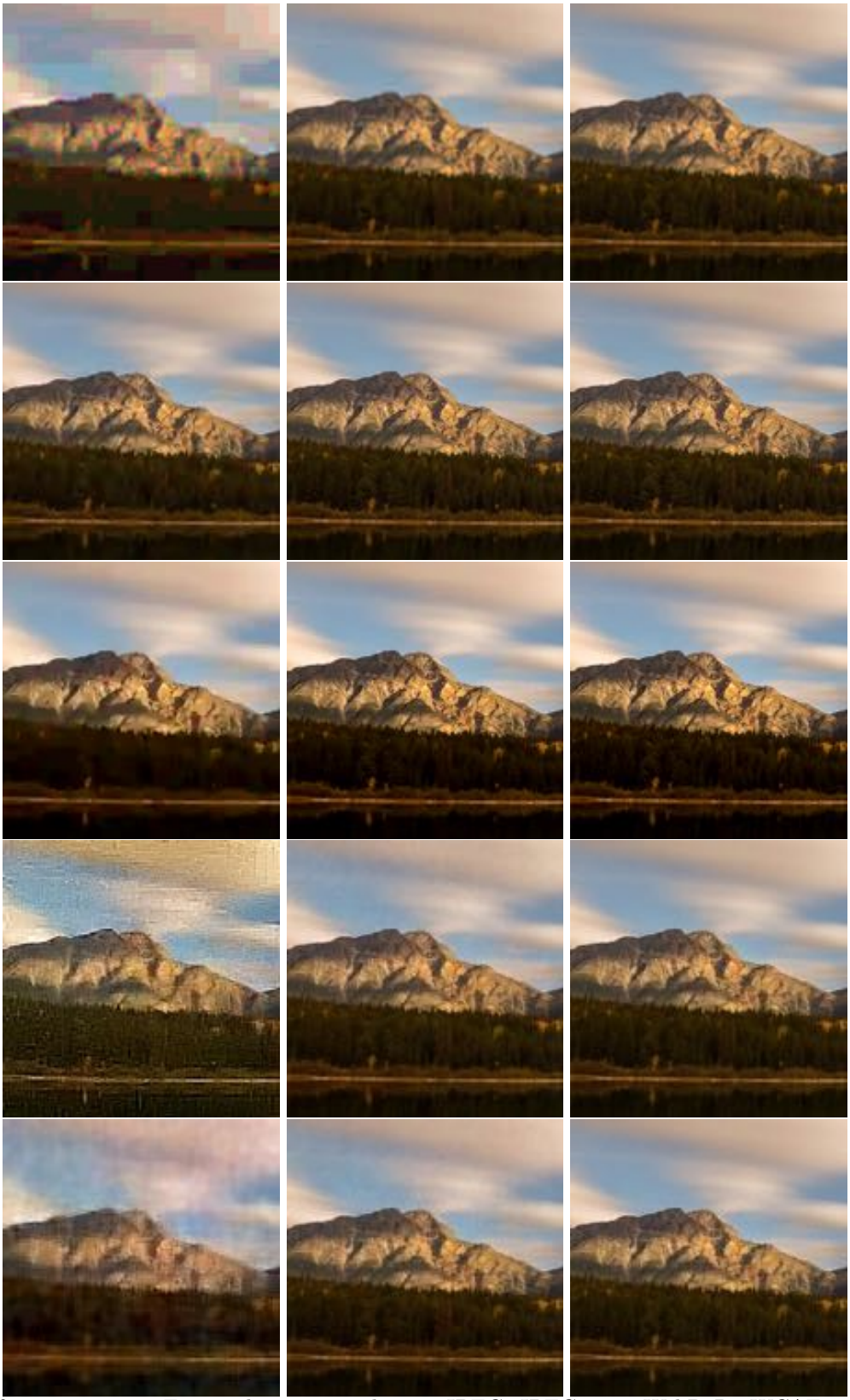


Figure 7: Reference image in Figure 5b compressed using JPEG, JPEG 2000, WebP, ResWCAE and ResMixW-CAE from top to bottom, at target bitrates 0.12, 0.50 and 1.00 bpp from left to right, respectively.

Figure 8: Reference image in Figure 5c compressed using JPEG, JPEG 2000, WebP, ResWCAE and ResMixW-CAE from top to bottom, at target bitrates 0.12, 0.50 and 1.00 bpp from left to right, respectively.

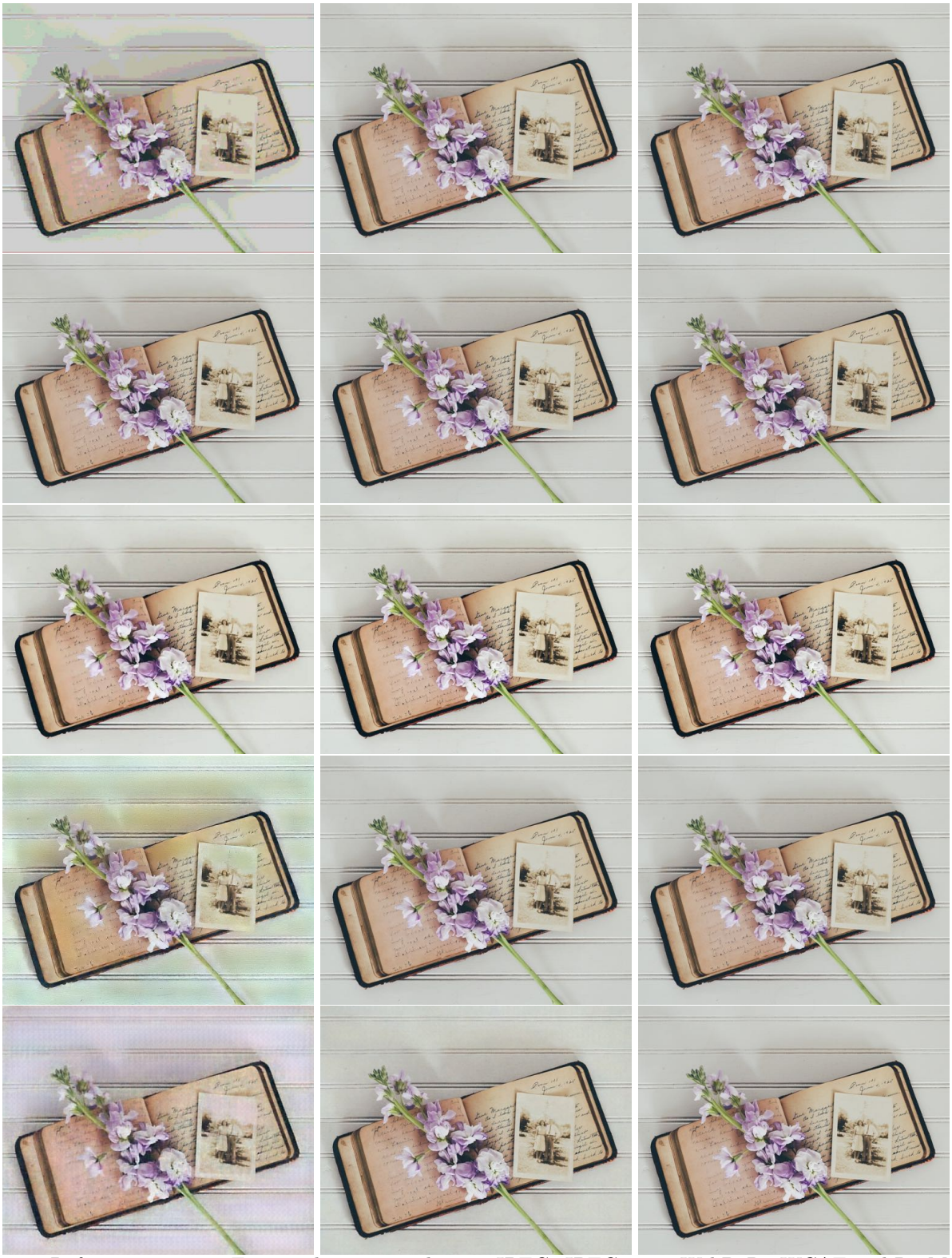


Figure 9: Reference image in Figure 5d compressed using JPEG, JPEG 2000, WebP, ResWCAE and ResMixW-CAE from top to bottom, at target bitrates 0.12, 0.50 and 1.00 bpp from left to right, respectively.

5. CONCLUSION

In this paper, two new end-to-end image compression architectures based on convolutional neural networks have been presented. The networks have been built as extensions to a previous model,[6] which uses 2D wavelet decomposition as a preprocessing step before training. ResWCAE and ResMixWCAE have deeper architectures than WCAE, employ residual connections and have larger output size at the analysis stage. ResMixWCAE also has varying kernel sizes along its branches processing wavelet coefficients at different scales. Results show that both models outperform JPEG compression, but are inferior to JPEG 2000 and WebP when compared using objective metrics. Subjective results indicate that ResWCAE and ResMixWCAE are able to preserve high frequency components, reduce blur and introduce no ringing or blocking artifacts. ResWCAE has more high frequency noise at lower bitrates, whereas ResMixWCAE suffers from more high frequency noise at higher bitrates. Future work involves testing different distributions of kernel sizes on all three scales and comparing the results with networks of similar architecture that do not use wavelet decomposition as a preprocessing step.

ACKNOWLEDGMENTS

This paper reports a research performed under the framework of project Digital Eye: Deep Learning Video Quality Assessment Technology, funded by The Swiss Commission for Technology and Innovation (CTI) under the grant 27403.1 PFES-ES.

REFERENCES

1. G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics* **38**(1), pp. xviii–xxxiv, 1992.
2. D. Taubman and M. Marcellin, *JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice*, vol. 642, Springer Science & Business Media, 2012.
3. "Webp compression study." https://developers.google.com/speed/webp/docs/webp$_s$*tudy.Accessed* : 2019 − 07 − 29.
4. G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology* **22**(12), pp. 1649–1668, 2012.
5. "Overview of jpeg xl." https://jpeg.org/jpegxl/index.html. Accessed: 2019-07-29.
6. P. Akyazi and T. Ebrahimi, "Learning-based image compression using convolutional autoencoder and wavelet decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.
7. "Workshop and challenge on learned image compression." https://www.compression.cc. Accessed: 2019-08-07.
8. J. Jiang, "Image compression with neural networks–a survey," *Signal processing: image Communication* **14**(9), pp. 737–760, 1999.
9. 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.
10. L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395* , 2017.
11. G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5306–5314, 2017.
12. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555* , 2014.
13. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
14. Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, **2**, pp. 1398–1402, Ieee, 2003.
15. J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704* , 2016.
16. Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," in *2018 Picture Coding Symposium (PCS)*, pp. 253–257, IEEE, 2018.
17. J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436* , 2018.
18. Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep residual learning for image compression," *arXiv preprint arXiv:1906.09731* , 2019.
19. J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," *arXiv preprint arXiv:1511.06281* , 2015.
20. "Kodak lossless true color image suite." http://r0k.us/graphics/kodak/. Accessed: 2019-08-07.
21. D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard," *IEEE Transactions on circuits and systems for video technology* **13**(7), pp. 620–636, 2003.
22. E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Advances in Neural Information Processing Systems*, pp. 1141–1151, 2017.
23. 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* , 2018.

24. M. Tan and Q. V. Le, "Mixnet: Mixed depthwise convolutional kernels," *arXiv preprint arXiv:1907.09595* , 2019.
25. C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal* **27**(3), pp. 379–423, 1948.
26. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86**(11), pp. 2278–2324, 1998.
27. Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.
28. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980* , 2014.
29. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing* **13**(4), pp. 600–612, 2004.
30. H. R. Sheikh and A. C. Bovik, "Image information and visual quality," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, **3**, pp. iii–709, IEEE, 2004.
31. "Vmaf - video multi-method assessment fusion." https://github.com/Netflix/vmaf. Accessed: 2019-08-07.