WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Design and Implementation of an Adaptive CNN-Based Image Compression System with Attention Mechanism and Rate-Distortion Optimization

Ranjana [a]*, Ankitha [a], Rekha [a], Bhavyagowda [b]

[a] *Student, Department of Electronics and Communication Engineering, Bangalore Institute of Technology, Bengaluru, Karnataka, India*
[b] *Project Guide, Department of Electronics and Communication Engineering, Bangalore Institute of Technology, Bengaluru, Karnataka, India*
*E-mail: ranjanag0250@gmail.com

ABSTRACT

The increasing use of high-resolution images in areas such as surveillance, medical diagnosis and satellite imaging has created a need for compression methods that reduce storage requirements without losing important visual information. Traditional codecs, including JPEG and JPEG2000, apply the same basic compression process across an image, even though some regions may be more important than others. This paper examines three CNN-based approaches that adapt compression according to image content. The first uses Class Activation Mapping (CAM) generated by a VGG-style classifier to guide a four-level variable-quality JPEG scheme. The second uses a trainable Squeeze-and-Excitation (SE) attention network, optimized with a rate-distortion loss that also considers local image complexity. The third is a learned neural codec built around an attention-based encoder-decoder and a differentiable entropy bottleneck, trained end-to-end using an explicit rate-distortion objective. All three systems were implemented using TensorFlow 2 / Keras and tested on an 18-class subset of Caltech-256 containing 1,734 training images and 180 test images, each resized to 128 × 128 pixels. Performance was assessed using PSNR, SSIM, bits per pixel (bpp) and file-size reduction. The CAM-guided baseline reduced file size by 20.0% on eight test images. The trained attention model achieved an average reduction of 20.9% across the full test set, with an SSIM selectivity ratio of 5.45×. For 20 sampled test images, the learned codec recorded an estimated rate of 0.153 bpp, compared with 0.583 bpp for JPEG at the lowest tested quality (q = 5). Its PSNR was 0.33 dB lower than JPEG (21.14 dB compared with 21.46 dB), and its SSIM was also lower (0.548 compared with 0.602). The attention model additionally generated a plausible importance map for an ultrasound image that was not used during training. These findings are preliminary: the study uses a relatively small dataset, and the test set was also used for monitoring during training.

Keywords: Image compression; convolutional neural networks; attention mechanism; rate-distortion optimization; class activation mapping; PSNR; SSIM

1. Introduction

Image compression is a key part of many digital applications. Smartphones, surveillance cameras, medical scanners and remote-sensing equipment continuously produce large volumes of image data. High-resolution images can take up considerable storage space and require more bandwidth when they are transferred. Efficient compression is therefore important for storing and sharing these images.

Conventional image formats such as JPEG and JPEG2000 rely on fixed mathematical transforms, including the Discrete Cosine Transform (DCT) and wavelet transforms. In general, they apply a similar quantization strategy throughout the image, without distinguishing between visually important areas and less significant background regions. This can result in bits being used on areas that contribute less to the overall content, while useful details may be affected by compression artifacts.

Convolutional Neural Networks (CNNs) learn visual features such as edges, textures and object structures directly from image data. These features can help identify the parts of an image where quality should be retained. Research in learned image compression has progressed from autoencoder-based codecs with entropy models (Ballé et al., 2018) to attention-based and context-adaptive techniques (Cheng et al., 2020; He et al., 2022; Zou et al., 2022), as well as generative methods (Mentzer et al., 2020). Reproducing some of these advanced systems, however, can require considerable computational resources. Also, the relationship between image saliency and the rate-distortion trade-off is not always learned directly; some methods still depend on manually selected rules or heuristics.

This work explores a CNN-guided image-compression pipeline through three stages. At each stage, more of the adaptation process is handled by trainable components rather than manually defined rules. The main contributions are as follows:

1. A CAM-guided baseline is implemented in TensorFlow 2 / Keras. It uses the activation map to assign different JPEG quality levels to different regions of an image.
2. A trainable Squeeze-and-Excitation (SE) attention network is developed. Its loss function combines rate-distortion estimates from JPEG with a term that accounts for local image complexity.
3. A learned encoder-decoder codec is developed with a differentiable entropy bottleneck and is trained end-to-end without using JPEG during training.
4. The three approaches are evaluated against standard JPEG using PSNR, SSIM, bits per pixel (bpp) and file-size reduction. A medical image is also used for a qualitative test of generalization, and the limitations of the experiments are considered.

2. Related Work

JPEG and JPEG2000 are traditional image codecs that use DCT or wavelet transforms together with manually designed quantization. Learned compression follows a different approach by training the analysis and synthesis transforms alongside an entropy model. Cheng et al. (2020), for example, combined attention modules with discretized Gaussian mixture likelihoods. He et al. (2022) proposed ELIC, which uses unevenly grouped space-channel contextual coding. Zou et al. (2022) demonstrated the use of window-based attention to retain image details, while Mentzer et al. (2020) investigated high-fidelity generative compression. Yasin and Abdulazeez (2021) reviewed deep-learning-based image-compression techniques. Building on this research, the present work follows a staged implementation, starting with a saliency-guided conventional codec and progressing towards a learned codec. The approaches are evaluated on a comparatively small dataset using limited computational resources.

3. Methodology

Three approaches were implemented and evaluated. Each one introduces a higher level of learning-based adaptation to the compression process.

3.1 Dataset and Preprocessing

The experiments use 18 selected classes from the Caltech-256 dataset: birdbath, blimp, boom-box, bowling-ball, computer-keyboard, conch, desk-globe, doorknob, fire-truck, helicopter, human-skeleton, hummingbird, ice-cream-cone, joystick, ladder, microscope, telephone-box and tennis-court. This subset contains 1,734 training images and 180 test images. All images were resized to 128 × 128 pixels, and their pixel values were normalized to the range [0, 1]. A small number of additional X-ray and ultrasound images were used only for a generalization check. A separate validation set was not created for this study. Instead, the test images were also used as validation data during training and to choose the classifier checkpoint with the lowest validation loss. This overlap may make the reported test performance appear more favorable than it would on a completely independent test set. This limitation is discussed further in Section 4.4.

3.2 Approach 1: CNN-Guided Baseline

The first approach begins by training a VGG-style classifier from scratch. The network has 13 convolutional layers, followed by Global Average Pooling and a softmax layer. It was trained using the Adam optimizer with a learning rate of 1e-4, a batch size of 32 and a maximum of 30 epochs. Random flipping, rotation and zoom were applied to augment the training data. The classifier achieved 37.2% test accuracy, compared with a chance level of 5.6%. A Class Activation Map (CAM) was then generated using the method proposed by Zhou et al. (2016). It is calculated from the final 8 × 8 convolutional feature maps (conv5_3), using the dense-layer weights corresponding to the predicted class. ReLU and max-normalization are applied, after which the map is resized to 128 × 128 pixels using bilinear interpolation. The pixel values are divided into four equal ranges and mapped to JPEG quality levels 15, 40, 70 and 95. The output image is created pixel by pixel by selecting the corresponding regions from the four JPEG-compressed versions of the input. For evaluation, a fixed CAM threshold of 0.5 is used to distinguish salient regions from the background.

3.3 Approach 2: Trainable Attention with Rate-Distortion Optimization

The second approach uses a separate CNN containing Squeeze-and-Excitation (SE) attention blocks (Hu et al., 2018) to produce an importance map at the input image resolution. In contrast to the baseline, the importance map is learned directly by the network rather than being obtained from a separately trained classifier. The network consists of three 3 × 3 convolutional layers with 32, 64 and 32 channels. Each layer is followed by an SE block, and a final one-channel sigmoid convolution generates the importance map. During training, each map value $m \in [0, 1]$ is converted into a virtual JPEG quality using $q = 10 + 85m$. The corresponding bitrate and distortion are estimated using cubic polynomial curves fitted to measured JPEG results. The measurements include mean bpp and 1 − SSIM for quality levels from 10 to 95. They were obtained from 30 training images, with five images selected from each of the first six classes. The resulting loss is

$$L = mean(C \times D) + \lambda R(1)$$

In this expression, D is the predicted distortion, represented by 1 − SSIM at the virtual JPEG quality, and R is the mean predicted bitrate in bits per pixel. C denotes local image complexity. It is calculated from the grayscale image using the variance in a 5 × 5 window and then normalized so that its mean is

one for each image. The rate-weighting factor is set to $\lambda = 0.15$, while the complexity map weights the distortion term. The network was trained with Adam at a learning rate of 1e-3, a batch size of 16 and for 20 epochs. In an initial run without the complexity term, the network produced an almost uniform importance map, and the loss showed little change over the 20 epochs. The complexity term was therefore added to encourage a more varied response. With this term and $\lambda = 0.05$, file size decreased by 13.1%. The value of λ was then raised to 0.15 to obtain a reduction closer to the 20.0% baseline result (Table 1). During evaluation, the predicted map is divided into the same four ranges as in the baseline, corresponding to JPEG quality levels 15, 40, 70 and 95. Thus, the continuous virtual quality used during training is replaced by four discrete quality levels during evaluation.

3.4 Approach 3: Learned Neural Codec

The third approach is an attention-based encoder-decoder CNN containing 576,343 parameters (Fig. 1). Its encoder has four convolutional layers, each using a 5 × 5 kernel and a stride of 2. The layers have 32, 64, 96 and 32 channels, respectively, and each is followed by an SE block. Starting with an input image of 128 × 128 × 3, the encoder produces an 8 × 8 × 32 latent representation containing 2,048 values. The decoder uses a similar arrangement, with four 5 × 5 transposed-convolution layers having 96, 64, 32 and 3 output channels. SE blocks follow the first three decoder layers, and the final layer uses a sigmoid activation. During training, quantization is approximated by adding uniform noise $U(-0.5, 0.5)$ to the latent values. At inference time, the values are rounded instead. The bitrate is estimated from the ideal code length of the latent representation under a learned zero-mean Gaussian model, which has one scale parameter for each latent channel. The estimate is calculated using the probability mass within each unit-width bin. The current implementation does not include an arithmetic coder, side information or a file header. The model is trained end-to-end with the loss $L = D + \lambda R$, where D is the mean squared error for pixel values in [0, 1], R is the bitrate in bits per pixel, and $\lambda = 0.01$. JPEG is not involved in training. The network was trained using Adam with a learning rate of 1e-3, a batch size of 16 and 25 epochs.

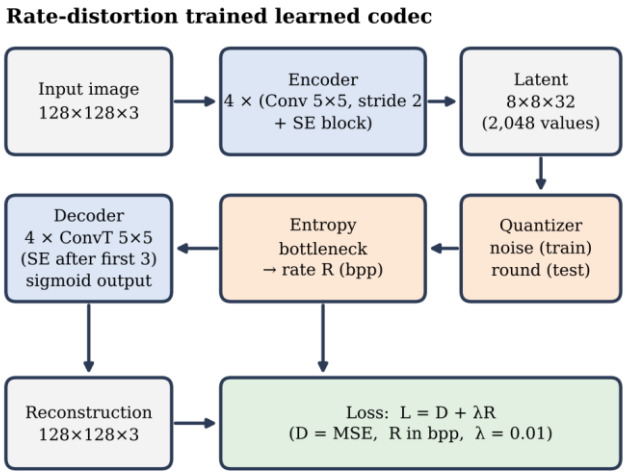


Fig. 1. Architecture of the learned neural codec: encoder, quantizer, entropy bottleneck, decoder and rate-distortion loss.

3.5 Implementation and Evaluation

The system was developed in Python using TensorFlow 2 / Keras and scikit-image. Google Colab and a T4 GPU were used to run the experiments. Performance was measured using PSNR, SSIM, bits per pixel and file-size reduction. For the first two approaches, the reference image is the original image saved as JPEG at quality 90. The adaptive output is assembled from the compressed regions and then saved again at quality 90 to measure its file size. PSNR and SSIM, however, are calculated on the assembled image before this final re-encoding. Region-based measurements use a salient mask, defined by importance-map values greater than 0.5; all remaining pixels are treated as background. Region SSIM is calculated by averaging the SSIM-map values within the relevant mask. Selectivity is measured as the background quality loss divided by the absolute quality loss in the salient region: $(Q_{uniform,bg} - Q_{adaptive,bg}) / |Q_{uniform,sal} - Q_{adaptive,sal}|$. Here, Q represents either PSNR or SSIM, calculated separately for the two regions. A value above 1 indicates that compression has affected the background more strongly than the salient region. The eight-image comparisons use randomly selected test images with seed 7. Although the baseline and trained-attention experiments used the same seed and class sequence, file-order differences between sessions meant that only one image was common to both samples (Section 4.2). The learned codec was evaluated on 20 randomly selected images using seed 3 and sampling with replacement. One image was selected twice, leaving 19 unique images in the sample.

4. Results and Discussion

4.1 CNN-Guided Baseline

On the eight test images used for this experiment, the CAM-guided baseline achieved a 20.0% reduction in file size. The loss in background quality was greater than the loss in the salient regions. Based on PSNR, the background reduction was 2.9 times larger (−9.43 dB compared with −3.29 dB); based on SSIM, it was 4.0 times larger (−0.095 compared with −0.024). On average, 36.0% of the pixels were marked as salient. This suggests that the approach retained more quality in the areas highlighted by the CAM. However, the CAM depends on the class predicted by the classifier, which achieved only 37.2% test accuracy. It may therefore highlight the wrong object in some images. For instance, one birdbath image was incorrectly classified as a fire-truck with 45.9% confidence.

4.2 Trained Attention with Rate-Distortion Optimization

The trained attention model was tested on another sample of eight images. The random seed and class sequence were kept the same as in the baseline experiment, but only one of the actual image files appeared in both samples. The trained model reduced file size by 20.9%, which is close to the baseline reduction of 20.0%. Its SSIM selectivity ratio was 6.43×, compared with 4.0× for the baseline, representing a numerical increase of about 61%. This difference needs to be interpreted carefully because the two methods were not tested on identical image samples, and the average salient-region coverage also differed: 16.4% for the trained model and 36.0% for the baseline. The trained model recorded a PSNR selectivity ratio of 2.4×, which was below the baseline value of 2.9×. The improvement is therefore evident in SSIM selectivity, but not in both quality measures (Fig. 2). When evaluated on all 180 test images, the trained model achieved an average file-size reduction of 20.9%, with a standard deviation of 7.4, an SSIM selectivity ratio of 5.45× and a PSNR selectivity ratio of 2.08×. The baseline was not evaluated across the full test set.

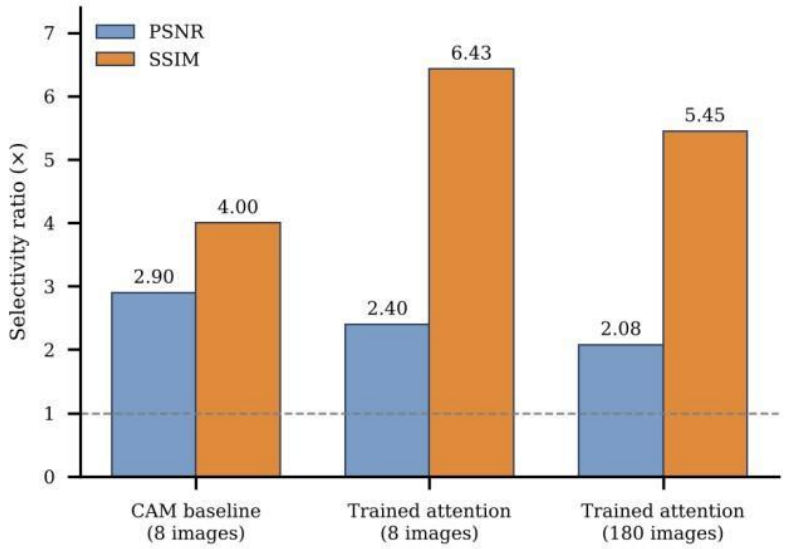


Fig. 2. Selectivity ratio (background quality loss divided by salient quality loss) for PSNR and SSIM. The dashed line marks 1×, meaning no selectivity. The two 8-image sets share a class sequence but only 1 of 8 images is identical.

Table 1 shows how different values of λ affected the trained attention model on the eight-image sample. A higher λ produced a larger file-size reduction and improved SSIM selectivity, but it also lowered PSNR selectivity. Since λ was chosen after the test-sample results had been examined, this value was tuned using the test results and should be interpreted accordingly.

Table 1. Rate weight λ in the trained attention model (8-image sample)

Model	**Size red.**	**PSNR sel.**	**SSIM sel.**
CAM baseline	20.0%	2.9×	4.0×
Trained attention, λ = 0.05	13.1%	3.5×	4.4×
Trained attention, λ = 0.15	20.9%	2.4×	6.4×

The trained attention model was also applied to an ultrasound image that had not been included in training. The network was used as it was, without any retraining (Fig. 3 and Table 2). This experiment gives a visual indication of how the model responds to an image from a different domain. The attention map marked 17.0% of the image pixels as important. With uniform JPEG compression at quality 90, the file size was 7,087 bytes, PSNR was 37.26 dB

and SSIM was 0.984. The adaptive output measured 6,100 bytes, with a PSNR of 30.28 dB and an SSIM of 0.937. The file-size reduction was therefore 13.9%. However, because this test involved only one image, it cannot establish how well the model performs on medical images in general.

Table 2. Single-image comparison (ultrasound sample)

Method	**Size (bytes)**	**PSNR (dB)**	**SSIM**
Uniform JPEG (q90)	7,087	37.26	0.984
Adaptive (trained attention)	6,100	30.28	0.937



(a) (b) (c)

Fig. 3. Ultrasound sample: (a) original image; (b) attention map; (c) compressed/decompressed output.

4.3 Learned Neural Codec

For the complete test set of 180 images, the trained codec produced a mean squared error of 0.0080, corresponding to approximately 21 dB, at 0.158 bpp. We carried out a more focused comparison using 20 sampled test images. For each image, JPEG quality was tested from q = 5 to q = 95 in increments of five, and the setting closest to the codec's estimated bitrate was selected. In all 20 cases, q = 5, the lowest quality level tested, was the closest match. Even at q = 5, JPEG had an average bitrate of 0.583 bpp, compared with 0.153 bpp for the learned codec. Since JPEG quality values below q = 5 were not examined, this was not a bitrate-matched comparison. At the tested settings, the codec achieved a PSNR of 21.14 dB, which was 0.33 dB below JPEG's 21.46 dB, while using less than one-third of the bits per pixel. Its SSIM was lower than JPEG's, at 0.548 compared with 0.602 (Table 3).

Two points are important when interpreting these results. First, the codec's bpp value is an estimate based on ideal entropy, whereas JPEG's bpp is calculated from the actual compressed file. This distinction matters for small 128 × 128 images because JPEG headers and quantization tables make up a relatively large share of the file. Second, the overall image quality for both methods is fairly low, at around 21 dB. The comparison also covers only one operating point. Full rate-distortion curves were not generated, and the codec was not compared with JPEG2000, WebP or BPG. The findings should therefore be limited to the settings tested here. It is also worth noting that the two quality metrics do not give the same result: the learned codec is close to JPEG in PSNR, but its SSIM is lower.

Table 3. Learned codec vs. JPEG at q = 5 (lowest tested quality), 20 sampled test images

Codec	**bpp**	**PSNR (dB)**	**SSIM**
JPEG (q = 5)	0.583	21.46	0.602
Learned neural codec	0.153	21.14	0.548

Table 4. Summary of the three approaches

Approach	**Key result (sample)**
CNN-guided baseline (CAM)	20.0% size reduction; background degrades 2.9× (PSNR) / 4.0× (SSIM) (8 images)
Trained attention + rate-distortion	20.9% size reduction; 6.43× SSIM selectivity on 8 images; 5.45× SSIM, 2.08× PSNR over 180 images
Learned neural codec	0.153 vs 0.583 bpp; PSNR 21.14 vs 21.46 dB; SSIM 0.548 vs 0.602 (vs JPEG q = 5, 20 images)

4.4 Discussion and Limitations

The results indicate that replacing the manually derived CAM map with an importance map learned through rate-distortion optimization can improve the selective allocation of SSIM quality, while keeping file-size reduction at a similar level. The same improvement was not observed for PSNR selectivity. The loss function gives more weight to areas with higher local image variance and does not use semantic labels. For that reason, the learned map is better interpreted as a measure of image-content complexity than as a detector of important objects. Region-based measurements are therefore useful when assessing this method. In the single-image ultrasound example, the adaptive output had a lower overall PSNR than uniform JPEG at q = 90 because the method deliberately reduced quality in background regions. Across the 180 test images, adaptive compression reduced file size by 20.9 ± 7.4%, and 17.7 ± 4.5% of pixels were marked as salient. Global PSNR fell from 35.62 ± 3.02 dB with uniform compression to 27.85 ± 2.73 dB with adaptive compression. Global SSIM changed from 0.966 ± 0.019 to 0.836 ± 0.056. In salient regions, PSNR changed from 32.86 ± 3.01 dB to 28.58 ± 2.71 dB, while SSIM changed from 0.989 ± 0.013 to 0.961 ± 0.023. In background regions, PSNR changed from 36.67 ± 3.15 dB to 27.79 ± 2.90 dB, and SSIM changed from 0.961 ± 0.021 to 0.810 ± 0.069. These results correspond to selectivity ratios of 2.08× for PSNR and 5.45× for SSIM.

The experiments have several limitations that should be considered when interpreting the results:

The experiments were carried out using a relatively small dataset of low-resolution images. The results may not transfer directly to larger images or to other application areas.

The test set was also used as validation data during training. It was involved in selecting the classifier checkpoint and influenced the choice of λ. This overlap may have resulted in more favorable test results than would be obtained using a fully independent test set.

Although the adaptive method reduced file size by 20.9%, global PSNR dropped by approximately 7.8 dB. The method was not compared with a uniform JPEG setting that produced a similar file size. Therefore, its benefit over a size-matched uniform JPEG encoder has not yet been demonstrated.

The adaptive image is assembled from four separately decoded JPEG images and is then re-encoded at quality 90 for measurement. It is not yet a single bitstream that can be decoded independently. The decoder-side information needed to identify the quality level assigned to each region has also not been implemented. In addition, training and evaluation use different quality representations: training uses continuous virtual quality values, whereas evaluation uses four discrete levels.

Only eight images were used to evaluate the CAM baseline. Just one image was common to both the baseline and trained-attention samples, so the comparison between the two methods is indicative rather than conclusive. The baseline is also affected by the classifier's limited test accuracy of 37.2%.

The medical-image generalization check was performed using a single ultrasound image. It is a qualitative example only and does not provide sufficient evidence to establish performance on medical images more broadly.

The learned codec was compared with JPEG at a single operating point using 20 sampled images, of which 19 were unique. A bitrate-matched comparison could not be made because q = 5 was the lowest JPEG quality tested. Furthermore, the codec's bpp value is an entropy estimate rather than the size of an actually encoded file, and its SSIM was lower than that of JPEG.

5. Conclusions

This study evaluated three CNN-based image-compression approaches: a CAM-guided JPEG baseline, a trainable SE-attention model optimized with a rate-distortion loss, and a learned neural codec. The trained attention model achieved a file-size reduction similar to that of the baseline, at 20.9% compared with 20.0%. It also produced a higher SSIM selectivity ratio in the sampled comparison, although its PSNR selectivity was lower. The model generated a plausible importance map for an ultrasound image that was not part of the training data, without requiring retraining. However, this observation is based on only one example. The learned neural codec recorded a PSNR close to JPEG, with a difference of 0.33 dB, while using less than one-third of JPEG's bitrate at q = 5 on the tested small images. Its SSIM was lower, and the comparison covered only one operating point. Future work should evaluate the methods on larger datasets and higher-resolution images, use a separate validation set, and compare the approaches at matched file sizes. Further experiments should also include JPEG quality settings below 5, complete rate-distortion curves and comparisons with other codecs. Other possible improvements include implementing actual entropy coding for the learned latent representation, adopting learned context-based entropy models, exploring perceptual and task-aware loss functions, and carrying out clinical validation using medical images.

Acknowledgements

The authors sincerely thank the Department of Electronics and Communication Engineering, Bangalore Institute of Technology, and the project guide for their guidance and support during this work. The authors also acknowledge the open-source tools TensorFlow, Keras and scikit-image, which were used in the implementation and evaluation of the proposed systems.

References

Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N., 2018. Variational image compression with a scale hyperprior, *Proc. International Conference on Learning Representations (ICLR)*.

Cheng, Z., Sun, H., Takeuchi, M., Katto, J., 2020. Learned image compression with discretized Gaussian mixture likelihoods and attention modules, *Proc. IEEE/CVF CVPR*, pp. 7939–7948.

He, D., Yang, Z., Peng, W., Ma, R., Qin, H., Wang, Y., 2022. ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding, *Proc. IEEE/CVF CVPR*, pp. 5718–5727.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, *Proc. IEEE CVPR*.

Mentzer, F., Toderici, G., Tschannen, M., Agustsson, E., 2020. High-fidelity generative image compression, *Advances in Neural Information Processing Systems (NeurIPS)* 33.

Yasin, H.M., Abdulazeez, A.M., 2021. Image compression based on deep learning: A review, *Asian Journal of Research in Computer Science* 8(1), pp. 62–76.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization, *Proc. IEEE CVPR*.

Zou, R., Song, C., Zhang, Z., 2022. The devil is in the details: Window-based attention for image compression, *Proc. IEEE/CVF CVPR*, pp. 17492–17501.