CompressNet: Generative Compression at Extremely Low Bitrates

Shubham Dash*, Giridharan Kumaravelu*, Vijayakrishna Naganoor*,
Suraj Kiran Raman*, Aditya Ramesh*, Honglak Lee
University of Michigan, Ann Arbor

Abstract

Compressing images at extremely low bitrates (< 0.1 bpp) has always been a challenging task as the quality of reconstruction significantly reduces due to the strongly imposing constraint on the number of bits allocated for the compressed data. With the increasing need to transfer large amounts of images with limited bandwidth, compressing images to very low sizes is a crucial task. However, the existing methods are not effective at extremely low bitrates. To address this need we propose a novel network called CompressNet which augments a Stacked Autoencoder with a Switch Prediction Network (SAE-SPN). This helps in the reconstruction of visually pleasing images at these low bitrates (< 0.1 bpp). We benchmark the performance of our proposed method on the Cityscapes dataset, evaluating over different metrics at very low bitrates showing that our method outperforms the other state-of-the-art. In particular, at a bitrate of 0.07, CompressNet achieves 22% *lower Perceptual Loss and* 55% *lower Frechet Inception Distance (FID) compared to the deep learning SOTA methods.*

1. Introduction

With the exponential growth of visual data-transfer, effective compression to extremely small scales is of paramount significance. In the case of images, classical image compression techniques such as JPEG [26], WebP [1], BPG [6] fail to give good quality reconstructions at low bitrates. However, lossy compression techniques using generative compression [2], [18], and [19] show promise in the reconstruction of aesthetically pleasing images at similar operating conditions.

Any lossy image compression scheme can be formulated as a rate-distortion optimization problem. In this framework, an analysis transform, $f : \mathbb{R}^N \to \mathbb{R}^M$, maps input data x to a vector z in latent space, and a synthesis transform, $g : \mathbb{R}^M \to \mathbb{R}^N$, transforms z back into the image space. An autoencoder setup is used to achieve these.

*equal contribution

Most of the existing compression systems are optimized for distortion metrics such as peak signal-to-noise ratio (PSNR) or different variants of structural similarity (SSIM) (Wang *et al.*, 2003). Traditionally emphasis has been put on building hand-crafted codecs (encoder-decoder pairs for compression tasks), by making strong assumptions such as the codec applying linear transform, as has been done with JPEG and JPEG2000. This assumption has an inherent problem as it is inaccurate to assume that a linear codec can generalize to compress a wide variety of natural images.

For very low bitrates, traditional metrics lose their relevance as they favor pixel-wise preservation of local structure over preserving texture and global structure. Recent works by Patel *et.al* in [17], [16] and Blau *et.al* in [7] indicate the need for better perceptual metrics that evaluate the visual quality of the images, rather than evaluating the structural similarity as captured by the traditional metrics. For a compression task, the reconstructions are required to have high perceptual quality, and also resemble the original image closely. Training a system with adversarial losses in this scenario produces better results as it enables a better understanding of the global structure of the image. We integrate a Generative Adversarial Network (GAN) setup along with the autoencoder for achieving this task.

The effectiveness of an autoencoder variant, stacked-autoencoders that incorporate layer-wise loss for learning latent dimensions for supervised tasks has been proven to enhance reconstruction quality for image compression tasks [29] over traditional autoencoders. We incorporate a similar idea in CompressNet for enhancing the reconstruction quality at extremely low bitrates. In addition to stacked-autoencoders, Stacked What-Where Autoencoder(SWWAE) [30] models suggest the use of pooling switch information for improved data-reconstruction across the encoder-decoder architectures. However, incorporating the pooling switch information increases data overhead making it infeasible for image-compression tasks at very low bitrates. To incorporate SWWAE models for compression tasks with no additional data-overhead, we propose a network to predict pooling switches and use it along with the SAE-all architecture. This allows us to operate

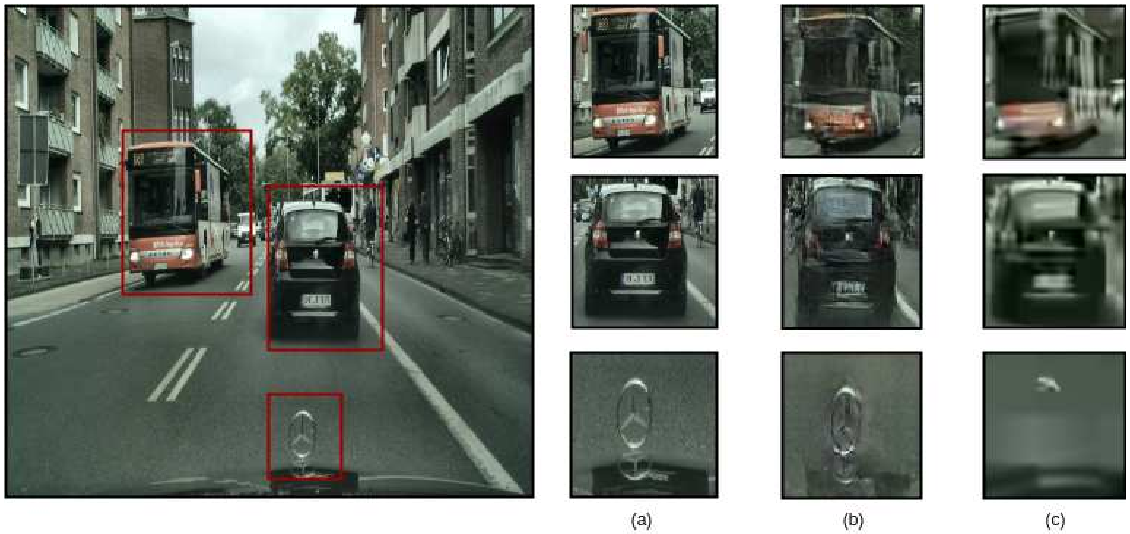


Figure 1: Original Image, a) Original image patches, b) SAE-SPN (ours) image patches, c) BPG image patches [6]

at extremely low bitrates adding very little computational overhead and shows comparable performance to SWWAE, which has proven to perform appreciably well for compression tasks.

In this paper, we propose three novel variants for extreme compression, summarized as:

- Stacked Autoencoder (SAE) based architecture with layer-wise loss
- Stacked What-Where Autoencoder (SWWAE) based architecture with layer-wise Loss
- Stacked Autoencoder with Switch Prediction Network (SAE-SPN) with layer-wise Loss

2. Literature Review

The classical approach to compression theory, mathematically formulated by Shannon's theory of communication [20], provides the fundamental basis that the coding theory is built on. Classical methods leverage explicit probabilistic modeling and feature extractions, effectively engineered for the task of image-compression [21], JPEG [26], and BPG [6]. Application of deep learning for image compression has emerged as an active area of research in the recent past. Incorporating autoencoder models into compression frameworks remains to be one of the most popular approaches amongst the deep learning techniques. Theis *et al.* [24], Balle *et al.*[5], Toderici *et al.*[25], Lee *et al.* [13] and Minnen *et al.* [15] have employed DNN architectures successfully for the task of image compression. Along with the autoencoders, GANs [9] have also been looked at as an alternative to the more traditional approaches such as JPEG [26] and BPG [6]. They tend to produce more aesthetically pleasing and accurate reconstructions. In this section, we specifically review image-compression frameworks that incorporate autoencoders and GANs.

2.1. Autoencoder

An autoencoder is a neural network that learns to reconstruct the input. It has a latent layer that describes a code used to represent the input to help reconstruct it back. Autoencoders have a constraint of not being able to be optimized directly due to the inherent non-differentiability of the compression loss. Mean-squared loss is generally used to measure the degree of distortion between the original and reconstructed images and is used to optimize the encoder-decoder network. Theis et al. [24] proposed a way to overcome this problem and have shown that minimal changes to the loss are sufficient to train deep autoencoders which are at par with JPEG 2000 in terms of the degree of compression making it suitable for compressing high-resolution images. Alexendre *et al.* (2018) [4] proposed using autoencoders along with residual blocks and skip connections to achieve lossy compression at low bitrates ($\backsim 0.15bpp$). However this approach suffers at extremely low bitrates ($\leq 0.1bpp$) because it optimizes for MS-SSIM, which emphasizes on pixel -level preservation of an image, leading to blurry reconstructions.

2.2. Generative Adversarial Network

GANs have been used for learning intractable distributions in an unsupervised manner. At very low bit-rates, compression networks based on reconstruction losses prove to be ineffective as they learn unimodal approximations of the real distribution. Fingscheidt *et al.* [14] showed using GAN architectures that traditional compression algorithms

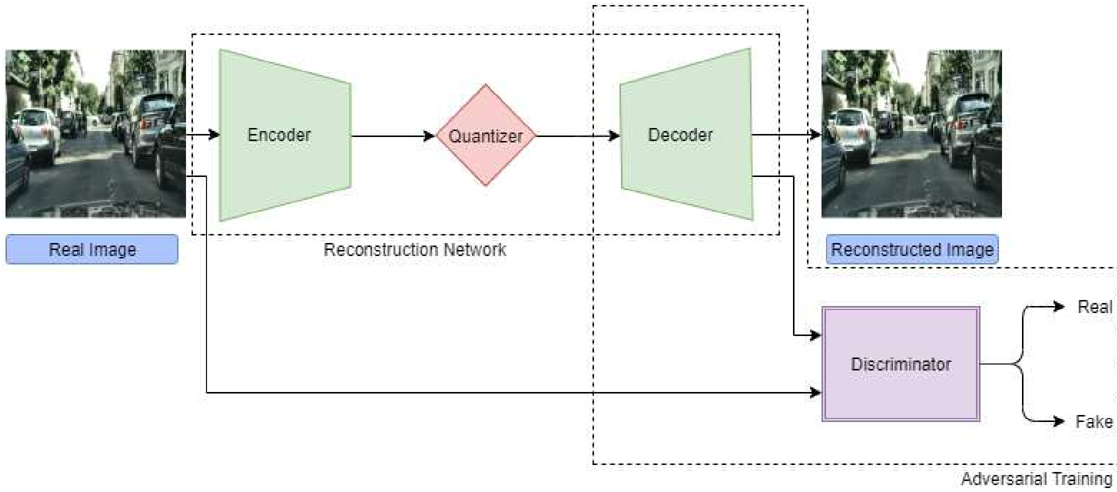


Figure 2: CompressNet Architecture

and techniques that use reconstruction loss to optimize for image compression lead to high PSNR and MS-SSIM but this does not necessarily translate to good perception functions, in this case, semantic segmentation. This necessitates the usage of GAN to capture the global structure and context of the image, enabling extreme learned compression. Given a data set X, GANs are to approximate its (unknown) distribution p_x through a generator G(z) that tries to map samples z from a fixed prior distribution p_z to the distribution p_x. This helps the generator to reconstruct the sharper images The generator G is trained in parallel with a discriminator D by searching for a saddle point of a mini-max objective. We have to take into consideration the reconstruction error and add the corresponding loss term $\mathbb{E}[d(x, g(D(G(z)))]$, which refers to the Vanilla GAN loss proposed in [9] to the total loss. The objective function now is

$$\min_G \max_D \mathbb{E}[f(D(x))] + \mathbb{E}[g(D(G(z)))] \\ + \lambda \mathbb{E}[d(x, g(D(G(z)))] \quad (1)$$

which implies minimising $\mathbb{L}_{GAN}$ over G.

$$\mathbb{L}_{GAN} := \max_D \mathbb{E}[f(D(x)] + \mathbb{E}[g(D(G(z))) \\ + \lambda \mathbb{E}[d(x, g(D(G(z)))] \quad (2)$$

In this work, we use $f(y)$ = $\log(y)$ and $g(y)$ = $\log(1 - y)$ used in Vanilla GAN proposed by Goodfellow et al.[9] which implies that we are finding the G minimizing the JS Divergence between the distribution of x and $G(z)$.

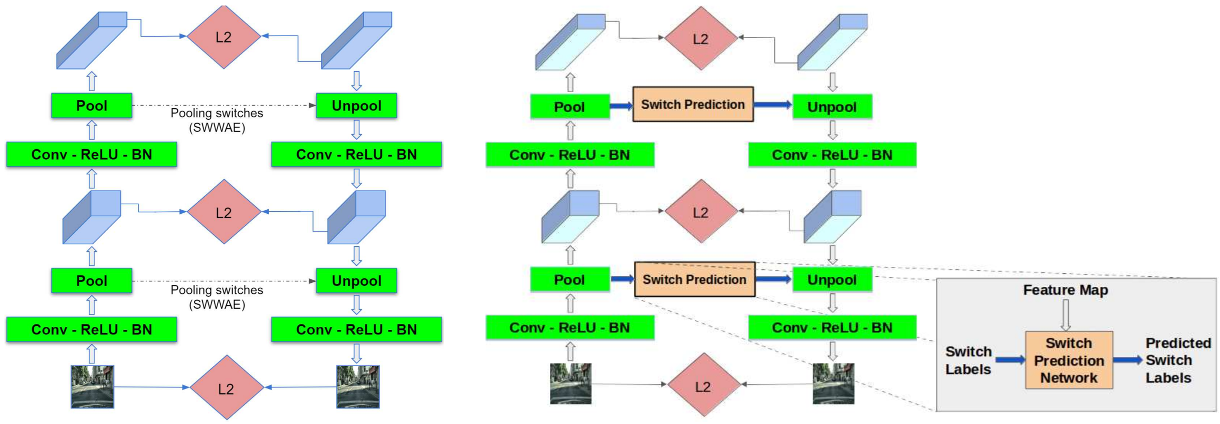
The effectiveness of Generative Compression can be attributed to the fact that the decoder is adversarially trained with a "paired" discriminator, similar to how a GAN is trained. It allows the decoder to learn the real distribution of the data and helps it generate visually pleasing reconstructions from a compressed latent representation.

Of late the image and video compression research community have increasingly shown a strong penchant towards the usage of GANs. The work by Santurkar *et al.* [19] is one of the early ones employing a GAN framework for image compression. Although they efficiently justify the potential of GANs, the work is more oriented towards representation learning on thumbnail images and not full resolution images. Rippel *et al.* in [18] proposed an adversarial framework for compression. It was primarily intended towards minimizing artifacts using an adversarial loss term, focusing on generating visually pleasing reconstructions. Agustsson *et al.* in [2] propose two networks for general and selective compression using conditional GANs. This is the current state of the art compression standard at extremely low bitrates. It has been inferred in [2] that the usage of conditional GANs is more pronounced in the case of selective object-based compression over general compression which is the current objective.

3. Method

The architecture for extreme learned compression (Figure 2) has been inspired by the recent work proposed by Agustsson *et al.* in [2], specifically, the architecture of the encoder E and the generator G proposed in Wang *et al.* [27]. The detailed explanation of both these architectures is explained below with the help of the diagrams.

The encoder takes in the image and converts it into a compressed feature space which is then passed through the Quantizer. Quantizer assigns a quantized value to each value in the compressed feature space based on the nearest quantization level, to obtain $\hat{w}$ a compressed and quantized representation. This forms the latent dimension from which the Decoder learns to reconstruct the original image back. This latent representation is then passed into the Generator



(a) SAE/SWWAE Architecture (b) SAE-SPN Architecture

Figure 3: Architecture Description

or Decoder, G which produces the reconstructed image $\hat{x}$. The discriminator, D is used for adversarial training which takes in this reconstructed image and the actual image. It predicts whether the given image is real or reconstructed. The discriminator follows the PatchGAN architecture [11]. A PatchGAN discriminator maps a 512 × 512 to a $N \times N$ array of outputs X, where each X_{ij} signifies whether the $patch_{ij}$ in the image is real or fake.

3.1. Approach

In order to improve the quality of reconstructions compared to existing to generative compression methods, we adopt three approaches as mentioned before. Each approach aims to modify the autoencoder setup in our model. These approaches are as follows:

3.1.1 Stacked Autoencoders

In this model, as shown in Figure 3a we compute layer-wise loss and add it to the final objective function and then optimize the entire network jointly. The layer-wise loss is calculated by taking the L^2 norm between the layer responses after every MaxPool - MaxUnpool operation in encoder-decoder architecture. This ensures that resulting reconstructions are as similar as possible even in the feature space and allows the encoded information to be propagated deeper into the encoding network, without much loss of information.

3.1.2 Stacked What-Where Autoencoders

Including pooling switch information infuses the decoder network with missing information. As a result during the reconstruction phase, the individual activations are placed at the location corresponding to the location where the maximum activation was observed during max-pooling in the encoding stage. However, this extra information comes at an additional cost of having to transmit these switch information from encoder to decoder. This increases the information overhead during compression and makes extreme compression infeasible since information transmission has to be minimized while keeping the reconstructions sharp.

3.1.3 Stacked Autoencoders with Switch Prediction Network

Even though the SWWAE architecture provides visually pleasing reconstructions the information overhead has to be eliminated to make it suitable for extreme compression. Incorporating pooling switch information seems to be the right direction to move ahead since the quality of the reconstructions obtained is significantly sharper. To retain the performance of the network in terms of perceptual quality along with traditional metrics like PSNR, $FSIM_C$ we predict the pooling switches using an auxiliary Switch Prediction Network (SPN) (Figure 3b). It is a convolutional neural network with a sigmoid activation function in the output. We assume a 2×2 max pool operation in the encoder, which maps a 4 element patch to a single value. 0 represents the top-left value in the 2×2 patch, 1 represents the top right, 2 represents the bottom left value and 3 represents the bottom right in the patch. For our experiments we have considered a 3×3 kernel to regress values in the range 0-1, then classes for predicting the max-pooling location are assigned as class 0 for values between 0-0.25, class 1 for values between 0.25-0.5, class 2 for values between 0.5-0.75, class 3 for values between 0.75-1. The functioning of this variant remains the same as SWWAE with the deployment of the switch prediction network in the decoder of the overall architecture.
Figures 3a and 3b explains how both the architectures are used for extreme compression. In both figures, 3a and 3b, the left side represents the encoder and right side represents

(a) Original Image (b) SAE-All (ours) @ 0.073 bpp (c) SWWAE (ours) @ 56.31 bpp

(d) BPG [6] @ 0.0726 bpp (e) SAE-SPN (ours) @ 0.073 bpp (f) SAE-All (ours) @ 0.036 bpp

Figure 4: Visual benchmarking of our proposed models with classical state-of-the-art method BPG[6]

the decoder. a) signifies the SAE and SWWAE architecture which is designed similarly with layerwise L_2 loss between each layer of encoder and decoder. Part b) of the figure describes SAE-SPN architecture. The salient difference is that instead of passing the pooling switch information like in SWWAE, we predict pooling switch based on decoder response. SWWAE uses pooling switch information to reconstruct the pixels at the exact location from where the max activation had taken place. Intuitively since we are predicting the pooling switches in SAE-SPN and no pooling switch information in SAE-All, the reconstructed pixels might not be always in the correct location of max activations giving a slightly inferior performance compared to SWWAE but has no information overhead. This makes it very feasible to be be used in extreme level compression.

3.2. Loss Function

The loss function used to optimize the entire pipeline consists of,

- Vanilla GAN loss function to optimize the generator and the discriminator, L_{GAN}
- Mean Squared Loss (MSE) to force the output reconstruction to be similar to the input image, L_{MSE}
- Perceptual Loss component to take care of the textural and feature similarity between the input and output images by minimizing the L^2 distance between the response from the 4^{th} convolutional layer of a pretrained Alexnet, $L_{perceptual}$
- SAE layer-wise loss the L^2 norm between the layer responses after every MaxPool - MaxUnpool operation in encoder-decoder architecture, L_{SAE}

$$\mathcal{L} = \min_{E,G} \mathcal{L}_{GAN} + \lambda_M \mathcal{L}_{MSE} + \lambda_p \mathcal{L}_{perceptual} + \lambda_S \mathcal{L}_{SAE} \tag{3}$$

where,

$$\mathcal{L}_{GAN} = E[\log D(x)] + E[\log(1 - D(G(z)))]$$
$$\mathcal{L}_{MSE} = \|x - \hat{x}\|_2$$
$$\mathcal{L}_{perceptual} = \|conv_4(x) - conv_4(\hat{x})\|_2$$

4. Experiments

4.1. Architecture, Losses, and Hyperparameters

The network architecture for our encoder and decoder/generator is based on the global generator network proposed by Wang et al. [28], in turn, based on the architecture proposed by Johnson et al.[12]

Encoder

Let `c7s1-k` denote a 7×7 Convolution-Instance Norm-ReLU layer with *k* filters and stride 1. `dk` denotes a 3×3

(a) Original Image (b) SAE-All (ours) @ 0.073 bpp (c) SAE-SPN (ours) @ 0.073 bpp

Figure 5: Comparison of CompressNet with SAE-All method; CompressNet reconstructs with greater detail

Convolution-Instance Norm-ReLU layer with *k* filters, and stride 2. We use reflection padding to reduce boundary artifacts. `Rk` denotes a residual block that contains two 3×3 convolutional layers with the same number of filters on both layers. `uk` denotes a 3×3 fractional-strided-Convolution-Instance Norm-ReLU layer with *k* filters, and stride $\frac{1}{2}$.

Architecture: `c7s1-60, d120, d240, d480, d960`

Decoder

Let `c3s1-960` denote a 3×3 Convolution-Instance Norm-ReLU layer with 960 filters and stride 1. `Rk` denotes a residual block that contains two 3×3 convolutional layers with the same number of filters on both layers. `uk` denotes a 3×3 fractional-strided-Convolution- Instance Norm-ReLU layer with *k* filters, and stride $\frac{1}{2}$.

Architecture: `c3s1-960, R960 X 9, u480, u240, u120, u60, c7s1-3`

Discriminator

Let `c4s2p1-k` denote a 4×4 Convolution-Leaky ReLU layer with k filters and stride value as 2 and padding value as 1 with k filters.

Architecture:`c4s2p1-64, c4s2p1-128, c4s2p1-192, c4s2p1-256, c4s2p1-512, c4s1p1-1`

We have also included a hard quantizer (non-differentiable), with L = 5 centers, $\mathbb{C}$ = {-2,-1,0,1,2}, to control the bitrate given by the expression, (Eq. 4). Additionally, we have incorporated sub-pixel convolutions with ICNR initialization [3], in place of the originally proposed convolution + upsampling in the decoder to get rid of checkerboard artifacts.

The encoder takes in an image of size H x W x 3 and returns a latent space dimension of H/16 x W/16 x $\mathbb{C}$. Hence the operating point characterized by bpp (Eq. 4) is directly related to the parameter $\mathbb{C}$. We experimented with the performance of our models, at C = {4,8} corresponding to 0.0363 bpp and 0.0726 bpp.

The encoder and the decoder/generator are trained with the Adam optimizer with a learning rate of 2e-3, coupled with a Learning Rate(LR) scheduler with a decay parameter of 0.5 for improved training. The discriminator is trained using the SGD optimizer with a learning rate of 2e-5.

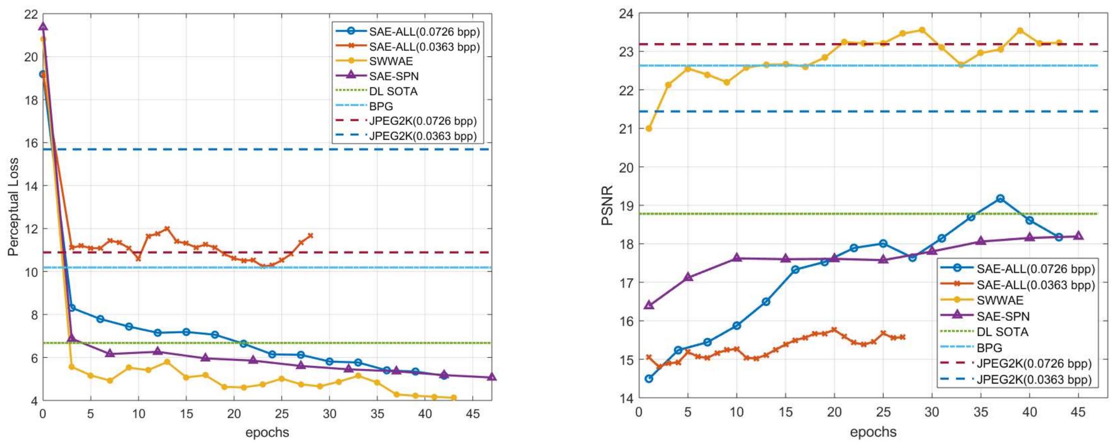
We are also predicting only the first layer switches for the encoder which is of dimension $256 \times 256 \times 60$, with the rest of the unpooling in the decoder done by Transposed Convolution. The intuition behind predicting the first level switches is while encoding the input onto a latent space, the first pool layer carries the most local information, making it essential to reconstructing the original image.

$$\mathrm{bpp} = \frac{\mathrm{H}/16 \times \mathrm{W}/16 \times \mathrm{C} \times \log_2 \mathrm{L}}{\mathrm{H} \times \mathrm{W}} \tag{4}$$

To obtain more visually pleasing reconstructions, we adopt $\mathcal{L}_{MSE}$ with a weight, λ_{MSE} = 1. Since we look to enhance the perceptual quality of the reconstructions, we incorporate $\mathcal{L}_{perceptual}$ based on AlexNet architecture proposed by [19] with a weight, λ_p = 5. In addition to the above losses, we incorporate the vanilla GAN loss $\mathcal{L}_{GAN}$, and the SAE layer loss $\mathcal{L}_{SAE}$ for sharper reconstructions, with weight λ_S = 1.

4.2. Datasets and Preprocessing steps

We train and evaluate our models on the Cityscapes dataset [8]. We enhance our models by including the CLIC (Challenge on Learned Image Compression) 2019 dataset to generalize better on the color information. The datasets were augmented to 18000 image patches of size 512×512 px generated with random crops and flips. Furthermore, Contrast Limited Adaptive Histogram Equaliza-



(a) Perceptual Loss with training

(b) PSNR Metric with training

Figure 6: Comparison of Perceptual and Traditional Metrics across models with training

tion (CLAHE) [31] was used to enhance the local contrast of these images before feeding them to the network.

4.3. Baselines

We benchmark all our compression models against traditional as well as deep learning-based state-of-the-art methods. BPG [6] is the current state-of-the-art engineered image compression codec, that outperforms the other recent codecs such as JPEG2000 [26] and WebP[1] in terms of PSNR. Specifically in the extreme-learned compression (bpp < 0.1) setting, generative compression proposed by Agussten *et al.* [2] is the current deep learning based state-of-the-art. Just for evaluation purposes, we use pre-trained weights of the same architecture [23] for comparison. Apart from the above state-of-the-art methods, we compare our models with other popular and common compression standards like JPEG2000 operated at similar bitrates, i.e 0.0726 bpp, and 0.0363 bpp.

4.4. Evaluation Metrics

We benchmark the performance of all our models with traditional metrics such as PSNR and SSIM. However, the primary focus is benchmarking based on perceptual quality.

In that regard, we evaluate the performance against perceptual loss, FSIM$_c$ and Fréchet Inception Distance (FID). Perceptual Loss is calculated as the L^2 distance between the response of the input and reconstructed image obtained after 4^{th} conv layer of the AlexNet. FSIM$_c$ is a measure that is based on the fact that the human visual system uses low-level features to interpret images.

A dimensionless quantity called phase congruence is used to calculate similarity between images. FID is a perceptual quality metric proposed by Heusel *et al.*[10] specifically, for evaluating the GAN synthesized images. FID uses the output features after the third pool layer of an inception [22] network, modelled using a multivariate Gaussian with mean μ and Σ. FID between the input dataset x, and reconstructed dataset g, is computed as,

$$\text{FID}(x, g) = ||\mu_x - \mu_g||_2^2 + \text{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{\frac{1}{2}}) \quad (5)$$

FID is a measure of how well the generated samples are approximating the real data distribution. Lower FID values signify the distance between the real and the generated data distribution is less and hence correlates with better image quality and diversity.

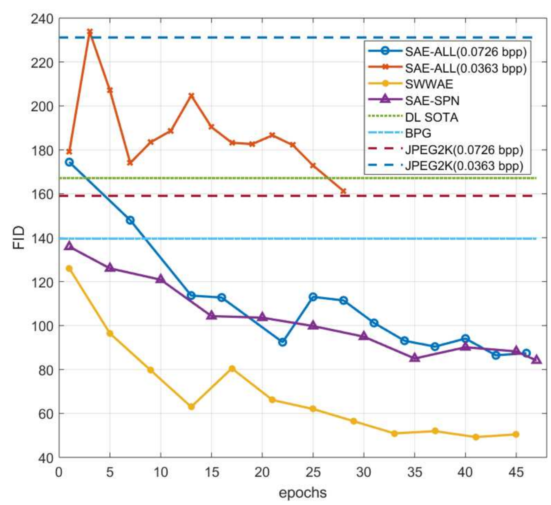
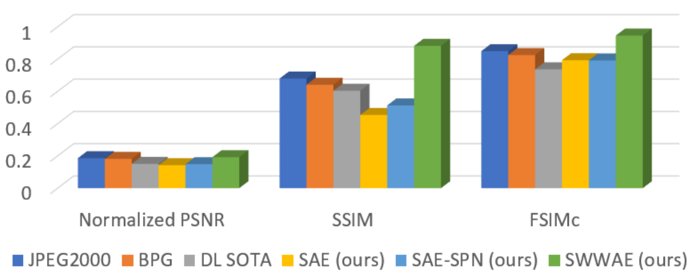


Figure 7: Comparison of FID across models with training

5. Results

To compare performance across different methods, we plotted different performance metrics discussed above, across epochs.



(a) Comparison of traditional metrics

	Bit Rate = **0.0726** bpp				
	SSIM	**PSNR**	**FSIM$_c$**	**PLoss**	**FID**
JPEG2K[26]	0.6793	23.1865	0.8491	10.89	159.05
BPG[6]	0.6411	22.6323	0.8240	10.19	139.58
DL SOTA[23]	0.6035	18.7794	0.7367	6.67	167.13
SAE (ours)	0.4536	17.7478	0.7932	5.15	87.44
SAE-SPN (ours)	0.5128	18.5084	0.7919	5.07	74.06
SWWAE (ours)	**0.8118**	**23.9258**	**0.9457**	**4.17**	**50.47**
	Bit Rate = 0.0363 bpp				
JPEG2K	0.6002	21.4389	0.7863	15.68	231.15
SAE (ours)	**0.3446**	**15.5972**	**0.6926**	**10.19**	**161.24**

(b) Comparison of perceptual metrics

Figure 8: Benchmarking our algorithm against competing algorithms

Plots in Fig. 6a is a representation of the relationship between perceptual loss and epochs for different methods. Following the intuition that since more information on pooling switches is being sent from the encoder to decoder, the quality of reconstruction achieved with SWWAE is significantly better than its counterparts like SAE-SPN and SAE-All, it also achieves the lowest perceptual loss implying better perceptual quality. SAE-All, CompressNet also shows comparable performance at 0.0726 bpp and far outperforms BPG and JPEG2000.

Plots in Fig. 6b is a representation of the relationship between PSNR and epochs for different methods. As observed, JPEG 2000 (at 0.0726 bpp) and BPG do appreciably well for PSNR metric, followed by SWWAE, SAE-All (at 0.0726 bpp) and SAE-All (at 0.0363 bpp). This is because traditional compression metrics optimize for PSNR but lose out on visual sharpness, as evident by reconstructions are shown in Fig. 4.

Plots presented in Fig. 7 describes the trends between FID and epochs for different methods. As discussed above, lower FID signifies better approximation to real data distribution and generates visually better-looking images. As expected SWWAE performs the best in this metric closely followed by CompressNet performance and SAE-All. This trend follows our intuition of SWWAE and CompressNet performing well on this metric due to the addition of pooling switch information. Traditional compression methods fall behind in this metric, as is evident from the plot. This is because traditional methods optimize for PSNR instead of a perceptual loss.

The bar plot aptly describes the performance of our methods against BPG and JPEG2000 for different metrics like Perceptual loss, SSIM, and FSIM$_c$. We have benchmarked the performance of our proposed methods against both the traditional and deep learning based state-of-the-art in fig 8.b. Although perceptual loss and FID are the primary metrics evaluating the visual quality of reconstructions, we have reported results against the traditional metrics as well. The methods we have proposed do comparably well on the traditional metrics and vastly outperform in terms of optimizing for perceptual quality of the image.

User study : To see if the perceptual quality and the FID metric are actually per the human perception, we conducted a small scale user study. In the survey, the original image was shown along with the reconstructed images obtained by 3 different methods CompressNet, BPG, and JPEG2K. 100 users from diverse backgrounds were asked to indicate their preference for each pair of reconstructions in the questionnaire. The percentage of preferred choice has been reported. This validates that CompressNet outperforms the traditional compression methods with superior perceptual quality.

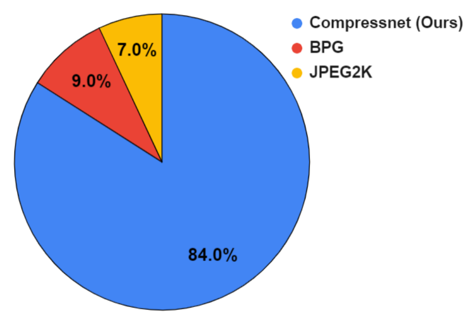


Figure 9: User study results indicating preference on image sharpness and quality across different methods

6. Conclusion

We have proposed and evaluated different GAN-based frameworks for extreme learned compression that significantly outperforms prior works for extremely low bitrates in terms of visual quality. Our proposed model, CompressNet (SAE-SPN) shows great promise for image compression as is evident by the results presented, where it performs comparably to traditional methods like JPEG2000 and BPG in terms of PSNR and FSIM$_c$, but is much superior to those methods when it comes to perceptual quality and FID. We believe learning compressed representations is a promising avenue to learn high-resolution generative models for multimodal data compression as well as adaptive image compression with wide-ranging applications.

References

[1] A new image format for the web. *https://developers.google.com/speed/webp/*.

[2] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool. Generative adversarial networks for extreme learned image compression. *CoRR*, abs/1804.02958, 2018.

[3] A. P. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi. Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. *CoRR*, abs/1707.02937, 2017.

[4] D. Alexandre, C. Chang, W. Peng, and H. Hang. An autoencoder-based learned image compressor: Description of challenge proposal by NCTU. *CoRR*, abs/1902.07385, 2019.

[5] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. *CoRR*, abs/1611.01704, 2016.

[6] F. Bellard. Bpg image format. *https://bellard.org/bpg/*.

[7] Y. Blau and T. Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. *CoRR*, abs/1901.07821, 2019.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[10] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[13] J. Lee, S. Cho, and S.-K. Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018.

[14] J. Löhdefink, A. Bär, N. M. Schmidt, F. Hüger, P. Schlicht, and T. Fingscheidt. GAN- vs. JPEG2000 image compression for distributed automotive perception: Higher peak SNR does not mean better semantic segmentation. *CoRR*, abs/1902.04311, 2019.

[15] D. Minnen, J. Ballé, and G. Toderici. Joint autoregressive and hierarchical priors for learned image compression. *CoRR*, abs/1809.02736, 2018.

[16] Y. Patel, S. Appalaraju, and R. Manmatha. Deep perceptual compression. *arXiv preprint arXiv:1907.08310*, 2019.

[17] Y. Patel, S. Appalaraju, and R. Manmatha. Human perceptual evaluations for image compression. *arXiv preprint arXiv:1908.04187*, 2019.

[18] O. Rippel and L. Bourdev. Real-time adaptive image compression. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2922–2930. JMLR. org, 2017.

[19] S. Santurkar, D. M. Budden, and N. Shavit. Generative compression. *CoRR*, abs/1703.01467, 2017.

[20] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 7 1948.

[21] C. R. Sims. Rate–distortion theory and human perception. *Cognition*, 152:181–198, 2016.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[23] J. Tan. Generative compression. *https://github.com/Justin-Tan/generative-compression*.

[24] L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. 03 2017.

[25] 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. *CoRR*, abs/1511.06085, 2015.

[26] G. K. Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[27] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, abs/1711.11585, 2017.

[28] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, abs/1711.11585, 2017.

[29] Y. Zhang, K. Lee, and H. Lee. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. *CoRR*, abs/1606.06582, 2016.

[30] J. J. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun. Stacked what-where auto-encoders. *CoRR*, abs/1506.02351, 2015.

[31] K. Zuiderveld. Graphics gems iv. chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.