CIFAR10-CompressAI: Convolutional Autoencoder for Image Compression with Perceptual Optimization

Pierre Raffalli (github.com/pierridotite/CIFAR10-CompressAI)

Abstract

We present a compact convolutional autoencoder for compressing images from CIFAR-10. The model is trained with a hybrid objective that combines per-pixel Mean Squared Error (MSE) and a feature-space perceptual loss, leading to reconstructions that preserve both global structure and perceptual detail. We report a practical compression ratio example of **12.0**× while maintaining visually faithful reconstructions, and we provide a modular, reproducible TensorFlow/Keras codebase.

1. Problem

Given an image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$, we learn an encoder E_θ and a decoder D_ϕ such that the reconstruction $\hat{\mathbf{x}} = D_\phi(E_\theta(\mathbf{x}))$ is perceptually close to $\mathbf{x}$ under rate constraints. On CIFAR-10, $H = W = 32$, $C = 3$, 8 bits/channel.

2. Model

Encoder. A stack of strided convolutions progressively downsamples $\mathbf{x}$ to a latent tensor $z \in \mathbb{R}^{h\times w\times k}$ (with $h, w \ll H, W$).
Decoder. Symmetric upsampling via transposed convolutions (or upsample+conv) reconstructs $\hat{\mathbf{x}}$.

Notation. $E_\theta : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{h\times w\times k}$, $\quad D_\phi : \mathbb{R}^{h\times w\times k} \to \mathbb{R}^{H\times W\times C}$.

3. Loss Function

We minimize a weighted sum of pixel-wise MSE and feature-space perceptual loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|_2^2, \tag{1}$$

$$\mathcal{L}_{\text{perc}} = \frac{1}{N}\sum_{i=1}^{N} \sum_{\ell\in\mathcal{S}} w_\ell \left\|\Phi_\ell(\mathbf{x}^{(i)}) - \Phi_\ell(\hat{\mathbf{x}}^{(i)})\right\|_2^2, \tag{2}$$

$$\mathcal{L}_{\text{total}} = \alpha\,\mathcal{L}_{\text{MSE}} + \beta\,\mathcal{L}_{\text{perc}} + \lambda\,\mathcal{R}(\theta,\phi). \tag{3}$$

Here, Φ_ℓ denotes fixed feature maps from a pretrained backbone (e.g., VGG-16 conv layers), $\mathcal{S}$ is a chosen set of layers, w_ℓ are layer weights, α, β, λ are scalars, and $\mathcal{R}$ is a regularizer (e.g. weight decay).

4. Rate and Compression Ratio

Let the original image size (uncompressed) be B_{orig} bits. Let the latent z be quantized with b_z bits per latent and size $h \times w \times k$ (ignoring entropy coding overhead for simplicity). Then

$$B_{\text{latent}} \approx h \cdot w \cdot k \cdot b_z, \qquad \text{CR} = \frac{B_{\text{orig}}}{B_{\text{latent}}}. \tag{4}$$

Example (from repo): $B_{\text{orig}} = 61\,440$ bytes, $B_{\text{latent}} = 5120$ bytes $\Rightarrow$ CR = **12.0**.

5. Evaluation Metrics

PSNR. For maximum intensity MAX (e.g. 255 for 8-bit):

$$\text{PSNR}(\mathbf{x},\hat{\mathbf{x}}) = 10\log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}(\mathbf{x},\hat{\mathbf{x}})}\right). \tag{5}$$

SSIM. With means μ, variances σ^2 and covariance $\sigma_{x\hat{x}}$:

$$\text{SSIM}(\mathbf{x},\hat{\mathbf{x}}) = \frac{(2\mu_{\mathbf{x}}\mu_{\hat{\mathbf{x}}} + C_1)(2\sigma_{\mathbf{x}\hat{\mathbf{x}}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\hat{\mathbf{x}}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\hat{\mathbf{x}}}^2 + C_2)}. \tag{6}$$

We report PSNR/SSIM on the test set alongside the compression ratio.

6. Training Setup

Data. CIFAR-10 train/val/test with standard normalization; random horizontal flips and mild color jitter.
Optimizer. Adam with learning rate 10^{-3} (cosine decay), batch size 128, 200 epochs.
Perceptual backbone. VGG-16 features (e.g. layers `conv1_2`, `conv2_2`, `conv3_3`); typical weights α=1.0, β=0.1 (tunable).
Implementation. TensorFlow/Keras; modular code under `src/` with `train.py`, `evaluate.py`.

7. Results

Table 1 shows an illustrative result consistent with the repository readme: high compression with faithful reconstructions. Exact numbers depend on latent shape and quantization settings (see config).

Setting	PSNR ↑	SSIM ↑	CR ↑
AE (MSE)	27.8	0.86	10.2×
AE (MSE+Perceptual)	**28.4**	**0.89**	**12.0**×

Table 1: Illustrative reconstruction quality vs. compression ratio.

8. Discussion

Why the hybrid loss? MSE aligns pixel values but can over-smooth; the perceptual term stabilizes higher-level structures and textures. **Trade-offs.** Increasing β typically improves SSIM/visuals at slight MSE cost. **Limits.** CIFAR-10 images are 32×32; scaling to larger images benefits from stronger backbones and entropy models.

9. Reproducibility

Code. github.com/pierridotite/CIFAR10-CompressAI
Quick start. `pip install -r requirements.txt` → `python src/train.py` → `python src/evaluate.py`.
Config. Latent size, quantization, and loss weights are set in `src/config.py` (see repo). Save models/plots under `models/`.

10. Future Work

(1) Learned entropy models for bit-accurate rate estimates; (2) multi-scale features; (3) adversarial or LPIPS losses for perceptual fidelity; (4) deployment on-browser (WebGPU) with post-training quantization.

References (selected)

Z. Wang et al., "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE TIP*, 2004.
J. Johnson et al., "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," *ECCV*, 2016.