Image Compression Benchmark

Ajay Joshi, Ritu Raut

Problem Statement

Image Compression

- How to represent an image using minimum amount of information?

Benchmark

- Present a comparison between classical and learning-based approaches to compress images

Motivation

Why Image Compression?

- Save storage space
- Save network bandwidth for transmission of images

Why is a benchmark useful?

- Presents different algorithms tailored to different set of requirements

Classical Algorithms

- **Lossless algorithms**: Can reconstruct each pixel in the original image exactly
  - Eg: PNG. Techniques used: Delta encoding, RLE Encoding, Huffman coding
- **Lossy algorithms**: Some loss in image quality occurs and have better compression ratios or bits per pixel (bpp)
  - Eg: JPEG, WebP, BPG:
  - JPEG: Techniques used - Color transforms, Discrete Cosine Transform (Fourier transform), quantization, Huffman coding

ML-Based approaches

- K-means clustering
  - Compression is limited as it replaces 3 bytes per pixel to 1 byte per pixel (cluster index)
- HiFiC (GAN-based)
  - Uses conditional GANs (upsampled image as condition) to generate the image, latents are the compressed representation
- Stable-diffusion
  - Idea can be understood as iterative denoising of the image, and the latents generated are the compressed representation

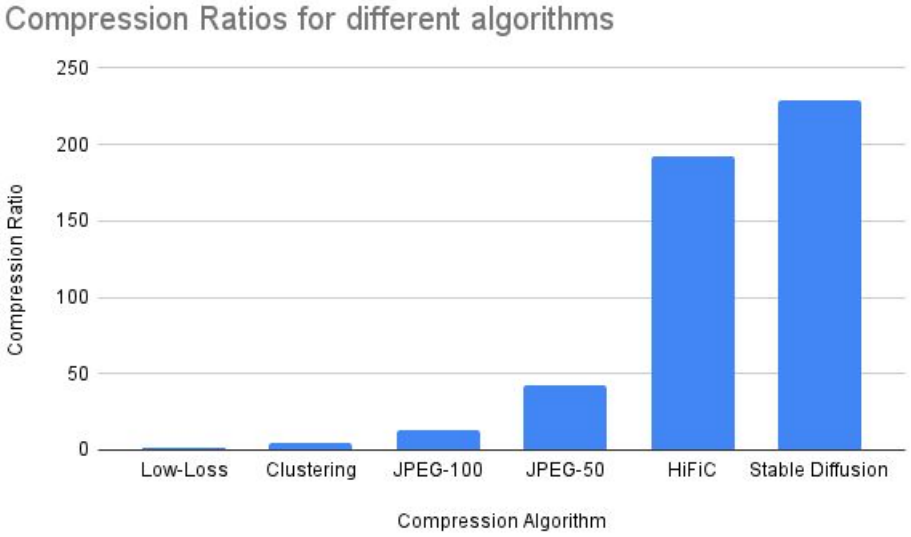
Image compression comparison tool

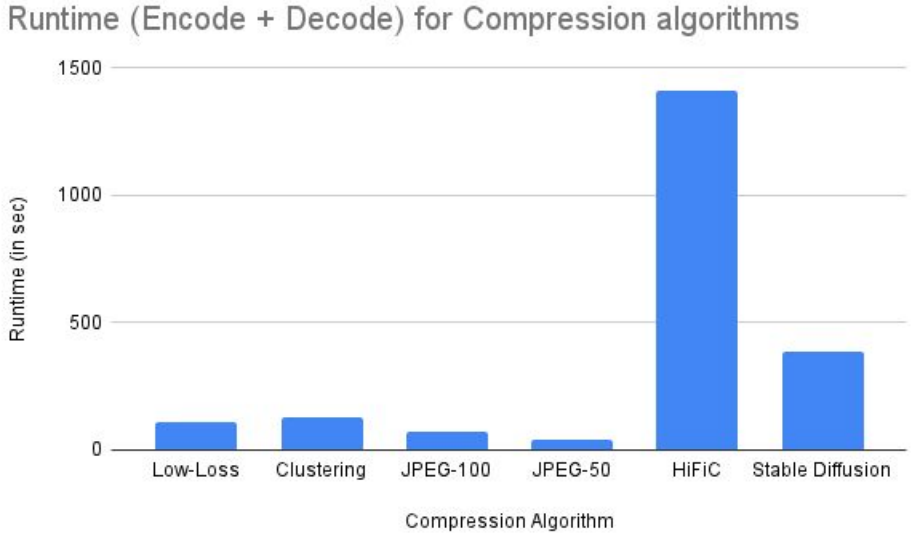
https://rituraut05.github.io/CVproject.github.io/

Evaluation

- **Dataset:** Image Compression Benchmark (link)
- **15** images of high quality and total size of **450MB**
- **Metrics:**
  - **Image quality**: PSNR, SSIM, Subjective evaluation
  - **Compressibility**: Compression ratio (Original size / Compressed size), Bits per pixel(bpp)
  - **Runtime**: Codec runtime for encoding/decoding

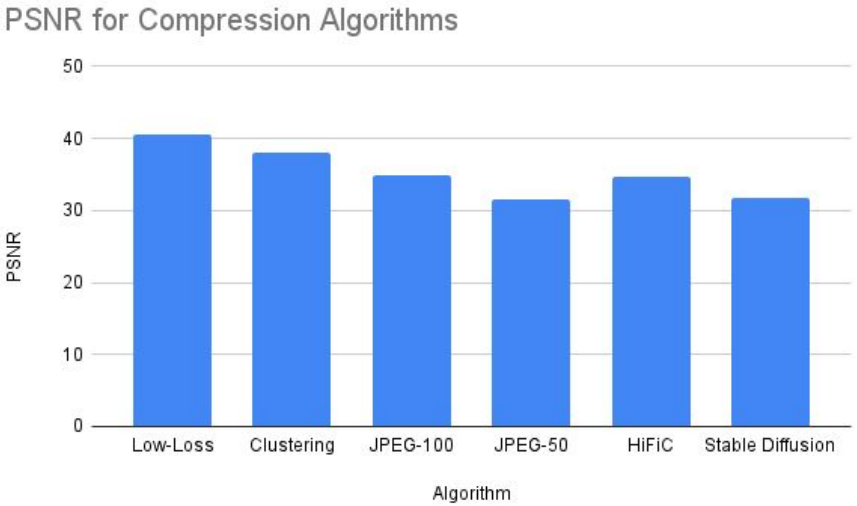
Results- Aggregate

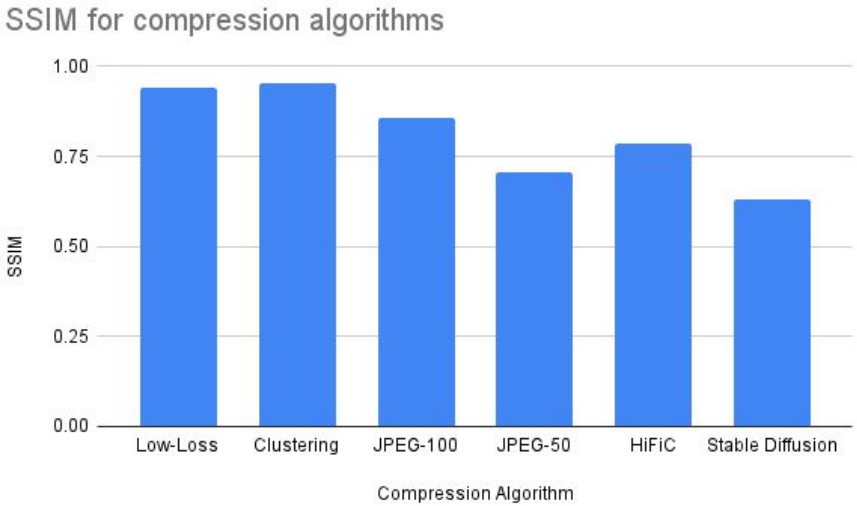




- Algorithms with the highest compression ratios have high codec runtime
- Use of specialized hardware is required(GPUs) for a better runtime for HiFiC and Stable Diffusion methods

Results- Aggregate

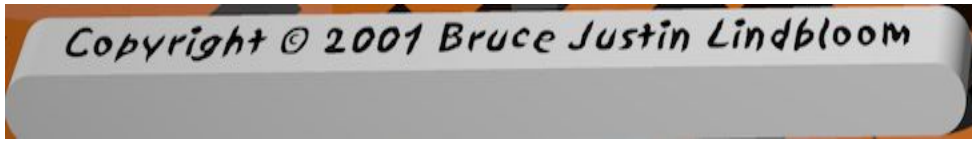




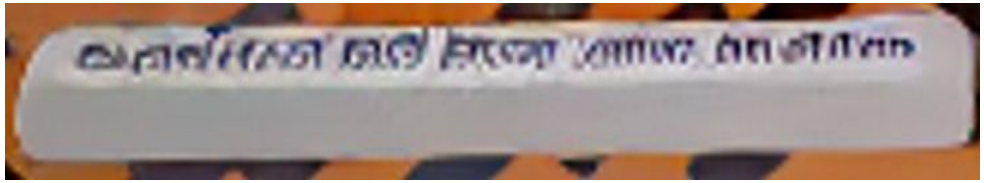
- PSNR and SSIM are not good indicator metrics of quality for generative methods, as they have lower score, but human evaluation shows a high-quality reconstruction

Results - Hallucination

Original



Stable Diffusion



Stable Diffusion hallucinates text in small(left image) and large fonts(right image).

Results - Hallucination

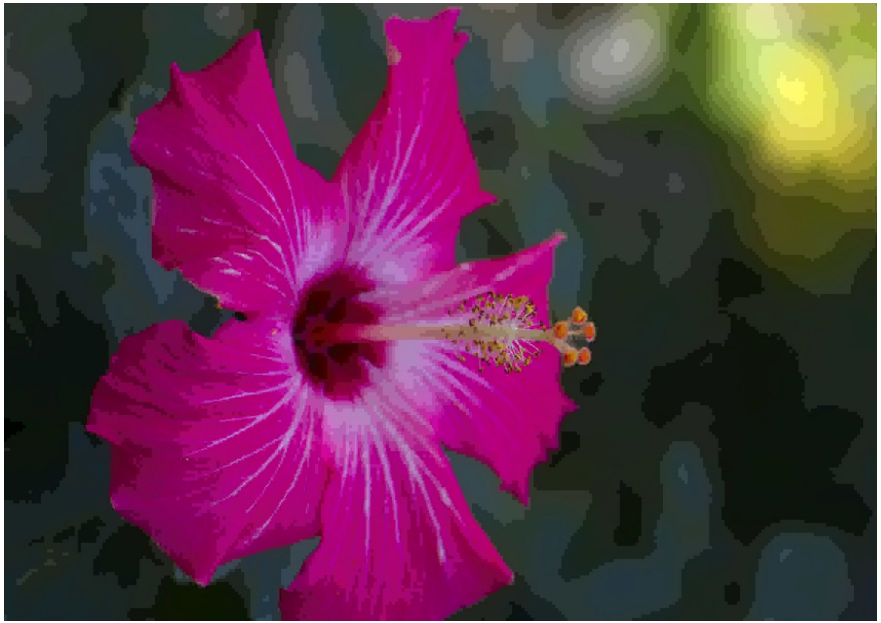
Original

Stable Diffusion

Stable diffusion also hallucinates some reasonable size details in images

Results

JPEG with quality factor 90



JPEG with quality factor 50

Patchiness in JPEG with quality level 50 due to high quantization

Results

K-Means with K = 200

Lossless compression

Patchiness in background due to limited clusters in K-means,

Results

HiFiC

Stable Diffusion

Both images are high quality, but stable diffusion hallucinates some details

Which compression algorithm should I use?

- Depends on use-case
- Trade-off between:
  - Size reduction
  - Quality of compressed image
  - Codec runtime

Algorithm Profiles

- High compression ratios (~150-200) + Good quality → High runtime (or require GPUs - higher cost) - **HiFiC, Stable Diff** (if some hallucination is fine)
- Medium compression ratios + Some loss in quality acceptable → Fast encode/decode possible - **JPEG, K-means** (with downsampling)
- High degree of fidelity to the original image + fast encode/decode → Lossless algorithms - **PNG**

Summary

- Implemented JPEG, PNG pipelines and associated image transforms
- Implemented ML-based methods (K-means) and used generative methods for compression (HiFiC, Stable Diffusion)
- High-performance multi-core implementations of JPEG, PNG, K-means
- Supportive for adaptive compression (JPEG quality factor with adaptive quantization, K-means - # of clusters + downsampling)
- Comparison on Image Compression Benchmark with detailed analysis of image quality, finding hallucinations
- Profiled each algorithm based on each metric considered

Future Work

- Add new methods/algorithms to the compression benchmark
- Add new metrics to evaluate algorithms better
- Identify compression techniques which are suitable for particular image domains

Thank You!