Context Aware GAN Image Compression

Evan Goldman
University of California, Riverside
egold018@ucr.edu

Hassan Rizwan
University of California, Riverside
hrizw002@ucr.edu

Figure 1: Original and compressed images of a sunflower. The compressed version takes almost 80% less storage while maintaining high quality

Abstract

Traditional image compression methods efficiently reduce image size but do not differentiate between relevant and irrelevant regions. In this work, we explore various approaches for detecting saliency in an image and leveraging this information to prioritize important regions during compression. Specifically, we examine three methods for identifying salient features: the spectral residual approach, gradient visualization in neural networks, and segmentation map confidence scores from a foundation model. Our findings indicate that the spectral residual method yields the best results. Upon downsampling the non-salient areas, we observe that in most cases, the image retains similar visual quality to human eyes, with the primary subjects remaining at full resolution. Building on this insight, we integrate the spectral residual method into a generative adversarial network (GAN) to enhance saliency-aware image compression. Traditional compression techniques uniformly allocate bits across an image, often neglecting the visual importance of different regions. Our project introduces a GAN-based, saliency-guided compression framework that distinguishes high-quality areas from less critical regions. By integrating adversarial training, the system refines the reconstruction of salient details while aggressively compressing non-essential parts. This selective bit allocation not only optimizes the rate-distortion trade-off but also maintains perceptual fidelity, ensuring that key features remain intact.

1 Introduction

Image compression is a technique used to reduce storage requirements while preserving image quality within acceptable limits. It is a critical component of data processing and computing, enabling

cost-effective storage, faster data transmission, and facilitating edge computing. The demand for high-resolution content in storage-constrained environments has driven ongoing research across various industries. For example, Adobe explores image compression to optimize data transmission and disk storage, while the aerospace industry investigates its applications for efficient storage and data transfer to edge computing devices such as satellites.

Image compression presents several challenges, including determining the most effective compression methods and balancing quality preservation with storage reduction. Traditional mathematical and data analysis techniques, such as Singular Value Decomposition (SVD) and Principal Component Analysis (PCA), offer efficient compression by reducing storage at the cost of some quality loss. However, these methods do not inherently differentiate between important and less important regions of an image, instead applying uniform compression across the entire image. As a result, users must manually define the desired compression level without considering the varying significance of different image regions.

In this work, we focus on prioritizing specific areas of an image—namely, its salient regions—during compression. We hypothesize that by allocating fewer bits to background or non-salient areas, we can achieve significant memory savings while preserving the meaningful parts of the image with minimal perceptual loss.

2 Related Works

Saliency with the Spectral Residual: Hou and Zhang (2007) propose a spectral residual approach to saliency detection by analyzing the log spectrum of an image and suppressing redundant spatial information. This method efficiently highlights visually significant regions without prior training. Their results demonstrate robust performance across various images, making it a widely used technique in saliency-based applications, including image compression and object detection.

Gradient Visualization: Selvaraju et al. (2017) introduce Grad-CAM, a gradient-based visualization technique that highlights important image regions influencing a neural network's predictions. By utilizing class-specific gradients in the final convolutional layers, Grad-CAM generates heatmaps to reveal salient features without modifying the network architecture. This method has been widely applied in interpretability and object detection.

Segmentation Confidence: The Segment Anything Model 2 (SAMv2) by Meta AI enhances segmentation by generating masks with confidence scores, in theory enabling saliency detection by giving high confidence to the subjects of the image. By leveraging these scores, SAMv2 identifies visually significant regions without prior training on specific datasets. Its output of segmentation masks along with confidence scores show promise towards segmentation confidence for salient features.

GAN image compression: The work of Liu, Yuchen, et al (2021) proposed a novel content-aware channel pruning and knowledge distillation techniques to compress unconditional GANs effectively. Their approach reduces StyleGAN2's FLOPs by 11x with minimal image quality loss while enhancing latent manifold disentanglement for better image editing. The work addresses the shortcomings of generic compression methods for GANs, paving the way for efficient deployment on edge devices.

3 Problem Formulation

Saliency of an image tells us what parts of the image are visually important to humans in identifying the contents of the image. We hypothesize that by leveraging the salient parts of the image, we can reduce the allocated bits to the non-salient images to greatly reduce the storage requirement of the image without degrading the important parts of the image. There are many different ways of finding saliency of an image ranging from classical image processing techniques, to machine learning methods. We did this project in two parts. First, we went over different methods of detecting saliency, performed some image compression between the 3, then we used those results alongside a GAN to perform more compression.

OpenCV has a direct saliency computation based on the spectral residual that we were able to leverage. We also investigated other ways to determine saliency. Grad-CAM is a method used for explainability in convolutional neural networks. It measures how much each pixel contributes to the

model's prediction. In theory, it is able to pick out the salient features in an image, separating it from the meaningless background. Another method we investigated is using SAM2's output scores. SAM2 gives a list of masks alongside a confidence score for each mask. We hypothesize that high mask scores would correlate with the subjects of the photo, as a way to separate out the background from the foreground.

Rapid growth in image sharing creates a pressing need for compression techniques to balance fidelity and storage efficiency. Traditional codecs apply uniform quantization across frames and achieve suboptimal results in key compression areas. We explore a pipeline which uses GANs alongside saliency-driven adaptive quantization. We use a saliency prediction module, which allows the system to identify regions of high importance and encodes them at higher quality, while compressing less crucial areas more aggressively. Moreover, a rate-distortion optimization mechanism ensures that the target bits per pixel (BPP) constraints are met without compromising crucial features. This way, perceptual quality is maintained while reducing file sizes. Real-world applications include medical imaging and remote sensing.

4 Experimental Results

Saliency Detection and Direct Compression

Alongside the three methods of determining saliency, we tried a compression approach that is broadly as follows. Detect salient areas Separate salient from non-salient areas Downsample the non-salient areas Save the salient image and the downsampled non salient image Then when reconstructing, Downscale salient image Add downscaled salient and non-salient images Upscale the new image Use new image to fill in missing areas in the salient image Note that we have to add the two downscaled images first, otherwise there will be stitching artifacts due to upscaling interpolation. We followed these steps for the three mentioned saliency detection methods. We also create a metric of theoretical file size reduction (Theoretical

Alongside the three methods of determining saliency, we tried a compression approach that is broadly as follows.

1. Detect salient areas
2. Separate salient from non-salient areas
3. Downsample the non-salient areas
4. Save the salient image and the downsampled non salient image

Then when reconstructing,

1. Downscale salient image
2. Add downscaled salient and non-salient images
3. Upscale the new image
4. Use new image to fill in missing areas in the salient image

Note that we have to add the two downscaled images first, otherwise there will be stitching artifacts due to upscaling interpolation.

We followed these steps for the three mentioned saliency detection methods. We also create a metric of theoretical file size reduction (Theoretical %). Currently, there is no file compression method that is compatible with our split images, so it is a rough estimate of how much less information is kept.

5 Compression with GANs

We also utilized GAN for compression and the pipeline is as follows.

5.1 Data Preprocessing and Saliency Computation

Images are loaded using OpenCV, converted from BGR to RGB, resized and normalized. Saliency maps are computed via OpenCV's static saliency algorithms. We used a combined method which

	MSE	PSNR	SSIM	Theoretical %	Time/200 im
OpenCV	45.23	31.58	0.9150	36.5	3s
Grad-CAM	47.49	31.36	0.9366	30.6	55s
SAM2	106.20	27.87	0.9661	24.3	2h

Table 1: Results on a dataset of 200 horse images. OpenCV had the best performance while having the shortest runtime. SAM2 had a catastrophic runtime of 36s per image.

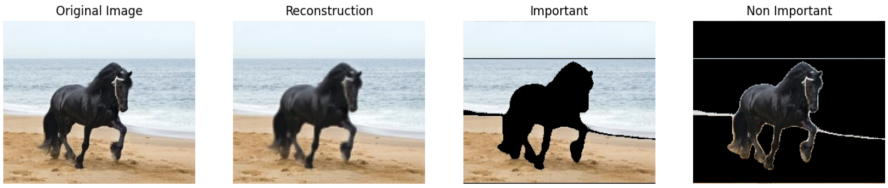

Figure 2: SAM2's results. It marks the background as salient and the subject as non-salient.

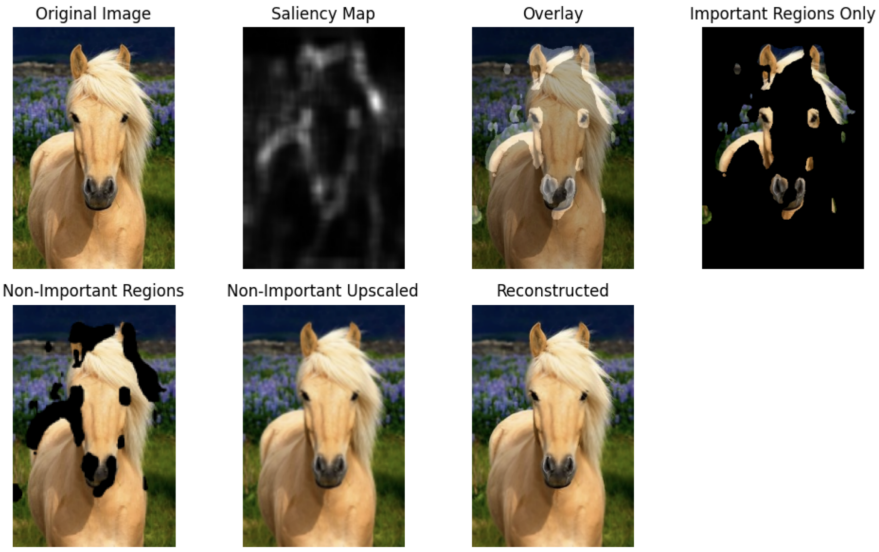

Figure 3: OpenCV's saliency results. Resulted in a 41% theoretical size reduction while maintaining quality of salient features.

calculates both spectral residual and fine-grained saliency maps and normalizes them. Adaptive saliency masks are generated using Otsu's thresholding and histogram-based adjustments. Smoothing is applied through bilateral filtering and Gaussian blurring to preserve edges and details.

5.2 Model Architecture

Dual Encoders: Two encoders are built—one for high-quality (HQ) regions with a latent dimension of 2× base and one for low-quality (LQ) regions with the base latent dimension. Both employ convolutional layers with skip connections; the HQ encoder integrates a custom self-attention layer at the 32×32 resolution to capture long-range dependencies.

Dual Generators: Each encoder has a corresponding generator composed of Conv2DTranspose layers. Skip connections from the encoders are concatenated at each upsampling stage to recover spatial details.

Latent Saliency Models: Separate latent saliency modules for HQ and LQ branches are implemented as dense networks with two hidden layers that output a saliency score per latent vector using sigmoid activation.

Adaptive Quantization: The custom AdaptiveQuantizationLayer takes the latent representation, the latent saliency score, and a quantization strength parameter. It applies soft, differentiable quantization by scaling the latent features via a factor derived from the quantization strength.

Rate-Distortion Optimizer: A specialized module processes the image, its saliency mask, and a target bits-per-pixel (BPP) value. It uses convolutional and global pooling layers to extract features, concatenates these with a normalized target BPP, and then outputs three parameters via lambda layers using tf.sigmoid: overall compression strength, an HQ/LQ threshold, and quantization strength.

5.3 Training Pipeline and Loss Functions

Adversarial Framework: The adaptive compression model is trained within a GAN framework using a discriminator network built with convolutional layers that outputs a probability via a sigmoid function. Reconstruction loss is computed as the mean squared error between the original and reconstructed images. Perceptual loss is obtained by comparing features extracted from a pre-trained VGG19 network (using multiple layers) after normalizing these features to zero mean and unit variance. L1 Loss is the mean absolute error used to further refine reconstruction quality. Latent Saliency Loss is the loss to encourage consistency between the predicted latent saliency scores and the pixel-level saliency masks—ensuring the HQ branch aligns with high saliency and the LQ branch with lower saliency.

5.4 Rate-Distortion and Bitrate Control Losses

A rate-distortion loss penalizes discrepancies between the computed overall compression parameter and a target compression level derived from the target BPP. An explicit bitrate control loss further ensures that the actual bits-per-pixel, computed from the latent dimensions (using 32 bits per float and the proportion of HQ vs. LQ regions), closely match the target BPP. Separate optimizers with tailored learning rates are applied to the main generator components, latent saliency modules, and the rate-distortion optimizer.

5.5 Testing and Evaluation (summary

The testing pipeline compresses images at various target BPPs (0.1, 1.0, 2.0). For each image, the pipeline computes the saliency map, generates an adaptive saliency mask, applies adaptive quantization, and reconstructs the image via the dual generator setup. Metrics such as PSNR, SSIM, compression ratio, and HQ region ratio are calculated. Visualization routines output the original image, the saliency map, the reconstructed image, and the bit allocation map, enabling detailed evaluation of compression quality and rate control.

6 Experimental Results

- Training and testing dataset: Both have 202 images each of horses and horse riders.
- Batch Size: The number of images processed in a single training step (16).
- Number of Epochs: Total passes through the training dataset (20).
- Learning Rate: Step size for gradient-based optimization (1e-4).
- Base Latent Dimension: Dimension of the latent space for low-quality encoding (512).
- High-Quality Latent Dimension: 2× the base dimension for HQ regions (1024).
- BPP Range: Range of target bits per pixel to train for ([0.1, 1.0, 2.0]).

BPP: refers to how many bits are used to encode each pixel in an image. A higher BPP means less compression (more data per pixel), while a lower BPP indicates more aggressive compression.

Target BPP: This is the desired or "goal" bits-per-pixel value that the compression process aims to achieve. The code typically tries to adjust its encoding strategy so that the output images meet (or

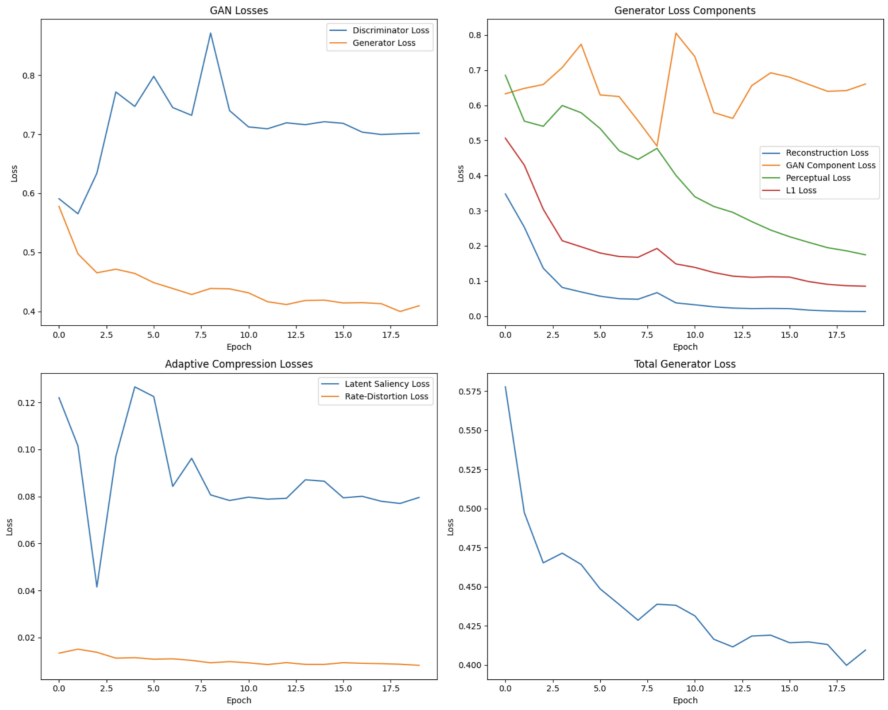

Figure 4: Training History

come close to) this specific BPP level, balancing file size and visual quality according to the user's requirements. After compression, the average bits per pixel in the image should be the target BPP, however some HQ regions would have received a higher BPP value, and to compensate some LQ regions would have received a lower BPP value.

6.1

GAN losses:

1. Discriminator Loss: Measures how effectively the discriminator can distinguish real samples from the generator's fake samples (lower loss indicates better discrimination).
2. Generator Loss: Reflects how well the generator fools the discriminator (lower loss means the generator's outputs are more "realistic" to the discriminator)
3. The first plot shows the training progression of both losses; as the discriminator becomes more accurate (its loss decreases), the generator adjusts its strategy to produce more convincing outputs (often causing its loss to fluctuate before stabilizing).

6.2 Generator Loss Components:

1. Reconstruction Loss and L1 Loss both measure how close the generated output is to the target image on a pixel or low-level basis. The Reconstruction Loss often refers to an MSE-like measure, while the L1 Loss directly sums the absolute differences between generated and ground-truth pixels.
2. GAN Component Loss ensures the generated images appear "realistic" by fooling a discriminator, while the Perceptual Loss compares high-level features (e.g., from a pretrained network) to maintain visually important details and structure.
3. The chart shows each loss decreasing at different rates as training progresses. The Reconstruction and L1 curves drop more steadily (improving pixel-level accuracy), while the GAN and Perceptual losses fluctuate as the model balances realism and high-level feature fidelity.

Figure 5: Horse image, saliency map, bit allocation map.

6.3 Adaptive Compression Losses:

1. Latent saliency loss focuses on preserving important or "salient" features in the latent space and puts higher weight on regions or features deemed critical for visual quality or semantic understanding, ensuring those areas are reconstructed with higher fidelity.
2. Rate distortion loss is a weighting parameter to control the trade-off between these two goals, ensuring that the model does not compress so aggressively that the visual or semantic quality is destroyed.
3. As training progresses, both losses move toward lower values—suggesting the model is improving at preserving essential features (latent saliency) while also managing the trade-off between compression efficiency and reconstruction quality (rate-distortion)

7 Acknowledgements

1. Spectral Residual from Hou and Zhang & OpenCV
2. Grad-CAM from Selvaraju et al.
3. Sam2 from FAIR (Facebook AI Research)
4. Dataset-Used (Kaggle: Pavan Sanagapati) Horses and Human training images used
5. AI Tools (Claude, ChatGPT)
6. OpenCV Saliency detection method used
7. TensorFlow, Keras, OpenCV, NumPy, Matplotlib, scikit-image, and SciPy

8 References

1. Hou, Xiaodi & Zhang, Liqing. (2007). Saliency Detection: A Spectral Residual Approach. IEEE Conference in Computer Vision and Pattern Recognition. 2007. 10.1109/CVPR.2007.383267.
2. Selvaraju, Ramprasaath R., et al. "Grad-CAM: Why did you say that?." arXiv preprint arXiv:1611.07450 (2016).
3. Ravi, Nikhila, et al. "Sam 2: Segment anything in images and videos." arXiv preprint arXiv:2408.00714 (2024).
4. Liu, Yuchen, et al. "Content-aware gan compression." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.