Optimal Image Compression through Integration of Deep Learning Architectures

Krish Siddhiwala

May 2025

1 Abstract

This work emphasizes a lightweight compression model that can be deployed in resource-limited scenarios such as mobile devices, drones, and IoT systems. Although conventional image compression methods such as JPEG and PNG are used, they lack adaptability and fail to leverage the capabilities of modern deep learning. In this paper, we propose a lightweight, end-to-end deep learning architecture for image compression, combining Convolutional Autoencoders (CAEs), Residual Blocks (ResBlocks), and Generative Adversarial Networks (GANs). This hybrid approach aims to balance compression efficiency with perceptual quality while remaining deployable for general-purpose tasks. This experiment conducts an ablation study to evaluate the contributions of each architectural component and benchmark performance using standard metrics such as Compression Ratio (CR), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM). The results highlight the effectiveness of the lightweight deep learning compression model, making it suitable for real-world edge applications.

2 Introduction

The exponential growth of visual data is reshaping the demands for data storage and visual data compression. For example, modern AI applications for mobile phones in low-bandwidth areas, drones in disaster recovery, and IoT devices in agriculture require lightweight compression models so that visual data can be compacted and stored efficiently. Recent studies estimate that over 80% of the data generated by edge devices come in the form of images or video [1, 8]. However, on-board memory constraints often prevent these data from being stored efficiently, leading to the loss of valuable information. Therefore, data storage becomes a struggle unless compression becomes more deployable with limited resources. To address these challenges, today's applications mostly rely on conventional methods such as JPEG and PNG. However, these methods are limited to predefined algorithms and do not take advantage of deep learning adaptive compression [18, 4].

Others have already taken steps in this direction with CAEs [2, 12, 15], which reduce images using encoder-decoder architectures. CAEs [3, 19] are a specialized neural network in image compression and reconstruction, incorporating the ability to learn complex data representations. By reducing images to low-dimensional latent spaces, autoencoders reduce data storage requirements. This paper explores a hybrid deep learning architecture for image compression that combines CAEs, ResBlocks, and GANs [6, 16, 9]. CAEs follow frameworks that compress and decompress images, ResBlocks help deepen networks and capture fine details, and GANs help enhance perceptual fidelity.

Deep learning models for image compression have been studied in the last couple years, and many of the techniques explored in this paper have been previously proposed. However, earlier papers have focused on specific application cases or assume high computational availability with large-scale models. The primary contribution of this paper is to demonstrate the

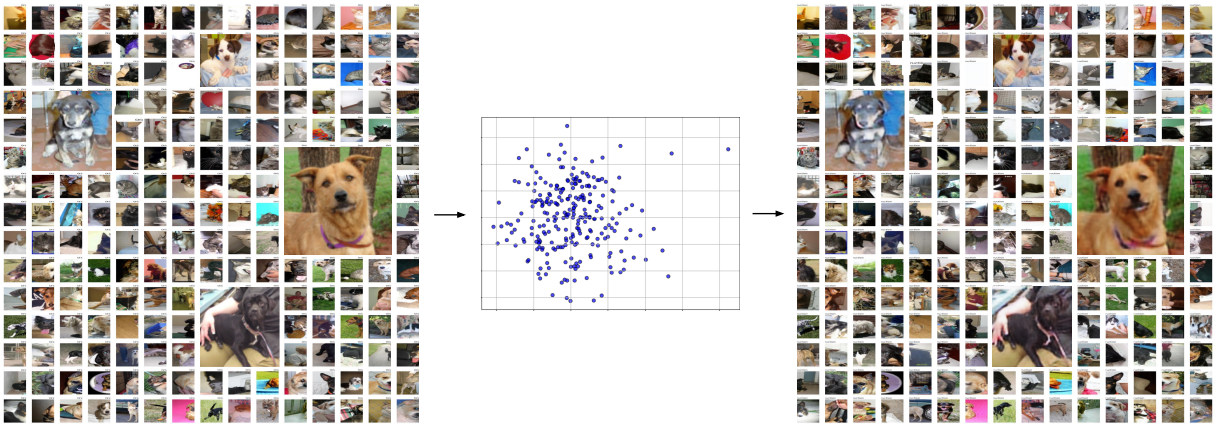

Figure 1: Visual overview of the compression and reconstruction output with images grids is shown. The images after compression are displayed with a latent space visualization graph. The original images are presented after compression and reconstruction through the proposed hybrid deep learning model. The images are from the Microsoft Kaggle Cats and Dogs dataset, which was also used to train the model. Four Specific images are enlarged to highlight the reconstruction quality. Full project page with examples is available at –.

effectiveness of a lightweight deep learning model towards producing reasonable results in image compression tasks. This approach allows for more deployable models that are suited for image compression while retaining perceptual fidelity. The second contribution is to isolate and analyze the effectiveness of individual architectural models through an ablation study. Their effectiveness is also explored through comparisons with common visual performance metrics such as CR, PSNR, and SSIM.

Together, these contributions offer an investigation toward data compression with reasonable reconstruction quality that is suitable for real-world, compute- and resource-limited deployments. Code is available at ——.

3 Related Works

CAEs Many others have used CAEs for image compression tasks due to their encoder-decoder architecture, as shown in Figure 2. This architecture ensures efficient compression while retaining the majority of key features of the input image [5].

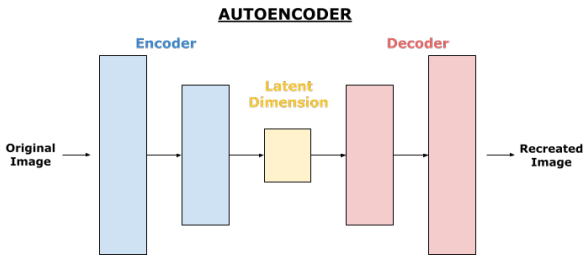

Figure 2: CAEs comprise two main components: the encoder and the decoder. The encoder use convolutional layers to compress the input image. The decoder utilizes transpose convolutions to upsample back to the original size. CAEs also have a middle section, known as the latent dimension or latent space. This section represents a lower-dimensional image state between the encoder and decoder.

Recent state-of-the-art image compression methods have adopted Transformer-based architectures [7, 14] for their efficient parameterization and ability

to capture contextual features, as well as UNet-style architectures [13], which leverage multiscale feature fusion to preserve spatial details in reconstruction. Although these advanced models offer some performance gains, CAEs remain a competitive and efficient option for many use cases, especially when balanced with perceptual and adversarial training losses.

CAE models are often trained with a foundational loss function, such as Mean Squared Error (MSE). It ensures that the output image closely approximates the original in terms of raw pixel values. However, when trained solely with reconstruction losses such as MSE, CAEs tend to produce overly smooth and blurry output, particularly in texture-rich regions.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{1}$$

Where x_i is the original image, $\hat{x}_i$ is the reconstructed image, and N is the number of pixels.

To address this limitation, previous work [11] has incorporated perceptual loss functions, such as Learned Perceptual Image Patch Similarity (LPIPS), which better aligns with human visual perception.

$$\text{LPIPS} = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\|w_l(l(x)_{h,w} - l(\hat{x}) \tag{2}$$

Including LPIPS in the loss formulation for image reconstruction helps shift the focus of the model from numerical accuracy to perceptual fidelity.

GANs GANs [9, 17, 20], as shown in Figure 3, also incorporate perceptual cues and have been adopted to improve reconstruction quality.

Recent literature has identified the benefits of adaptive loss weighting in multi-objective training. Dynamically balancing the components of the perceptual and adversarial loss components improves training stability and optimizes visual fidelity in a variety of content types. It also prevents the model from overfitting to one type of loss.

Hinge loss is also used in GAN models and has been shown to yield more stable GAN training than the binary cross-entropy formulation. It imposes a

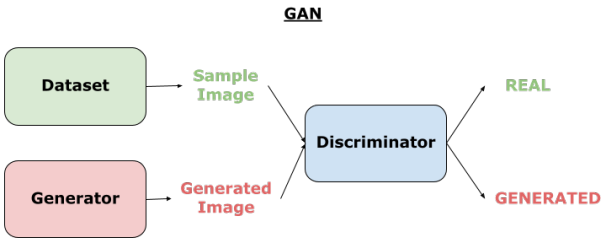

Figure 3: Schematic of a GAN for image compression. A GAN comprises two adversarial networks: the discriminator and the generator. The discriminator focuses on distinguishing real images from generated ones, and the generator seeks to improve the reconstruction quality of its generated images. Therefore, the GAN architecture employs the discriminator and generator to compete with each other to improve the generated image fidelity.

margin-based separation between real and fake images.

Discriminator

$$L_D = E[\max(0, 1-D(x))] + E[\max(0, 1+D(\hat{x}))] \tag{3}$$

Generator

$$L_G = -E[D(\hat{x})] \tag{4}$$

ResBlocks A major architectural improvement in recent models is the integration of ResBlocks [10], as shown in Figure 4. The Resblock architecture makes them well-suited for compression tasks where feature retention across network depth is critical.

This work builds on these insights directly by combining CAEs, ResBlocks, and GANs into a unified compression model. This paper uses CAE models, deepened with ResBlocks, GAN discriminators, and a loss function incorporating MSE, LPIPS, hinge loss, and adaptive weighting.

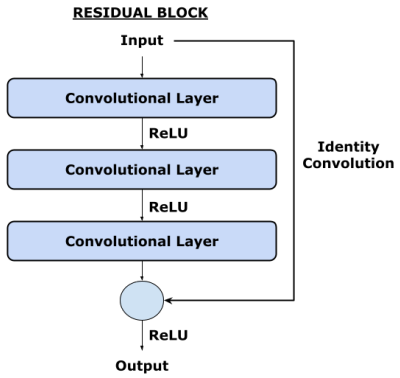

Figure 4: ResBlock architecture employs skip connections and deep convolutional layers, mitigating the issue of vanishing gradients, and ensuring gradient optimization. Each block includes two convolutional layers and a skip connection that adds the input to the output, preserving information across network depth and stabilizing training.

4 Methods

4.1 Dataset

This experiment used the Microsoft Kaggle Cats and Dogs dataset, which comprises 25,000 images, to train and evaluate the models. The images in this dataset, resized to 128x128, offer diversity in texture, shape, and contrast. Unlike domain-specific datasets, this dataset allows the evaluation of model generalization in everyday content.

4.2 Model

Generator Architecture The generator is structured as a symmetric autoencoder with residual connections, designed to capture and reconstruct complex image features while preserving spatial detail. The encoder processes an input image $x \in \mathbb{R}^{3\times128\times128}$ through an initial covolutional layer 3×3 with 32 output channels. It is followed by a sequence of residual blocks and downsampling operations. The encoder gradually increases the number of channels through the following progression.

$$3 \to 32 \to 64 \to 128 \to 256$$

The decoder reverses the operations of the encoder. It begins with the deepest feature map and applies a series of upsampling steps. Residual blocks are applied at each stage and the number of channels progressively decreases to produce the reconstructed image $\hat{x} \in \mathbb{R}^{3\times128\times128}$.

$$256 \to 128 \to 64 \to 32$$

Discriminator: PatchGAN To ensure perceptual quality in reconstructions, the model also uses a PatchGAN discriminator D, which classifies overlapping image patches rather than the entire image. This encourages the generator to produce realistic textures and details. The discriminator consists of five convolutional layers with increasing feature dimensions.

$$3 \to 64 \to 128 \to 256 \to 512 \to 1$$

A comprehensive pseudocode chart of the model training is shown. The complete architecture of the model is shown in Figure 5.

Table 1: Notations and definitions for variables indicated in training.

Symbol	Description
x	Input image
$\hat{x}$	Reconstructed image: $\hat{x} = G(E(x))$
z	Latent vector from encoder: $z = E(x)$
E, G, D	Encoder, Generator(decoder), Discriminator
θ_E, θ_G, θ_D	Parameters of E, G, D
$\mathcal{L}_{rec}$	Total reconstruction loss: L1 + LPIPS
$\mathcal{L}_{L1}$	L1 pixel loss: $\|x - \hat{x}\|_1$
$\mathcal{L}_{LPIPS}$	Perceptual loss: LPIPS(x, $\hat{x}$)
$\mathcal{L}_{GAN}$	Generator adversarial loss: $-\mathbb{E}[D(\hat{x})]$
$\mathcal{L}_G$	Generator loss: $\mathcal{L}_{rec} + \lambda_{GAN} \cdot w \cdot \mathcal{L}_{GAN}$
$\mathcal{L}_D$	Discriminator loss: hinge formulation
w	Adaptive weight: ratio of $\mathcal{L}_{rec}$ to $\mathcal{L}_{GAN}$
λ_{LPIPS}	Weight for LPIPS loss
λ_{GAN}	Weight for GAN loss
t	Training step
`START_DISC`	Step to start discriminator training
ϵ	Small constant to avoid division by zero

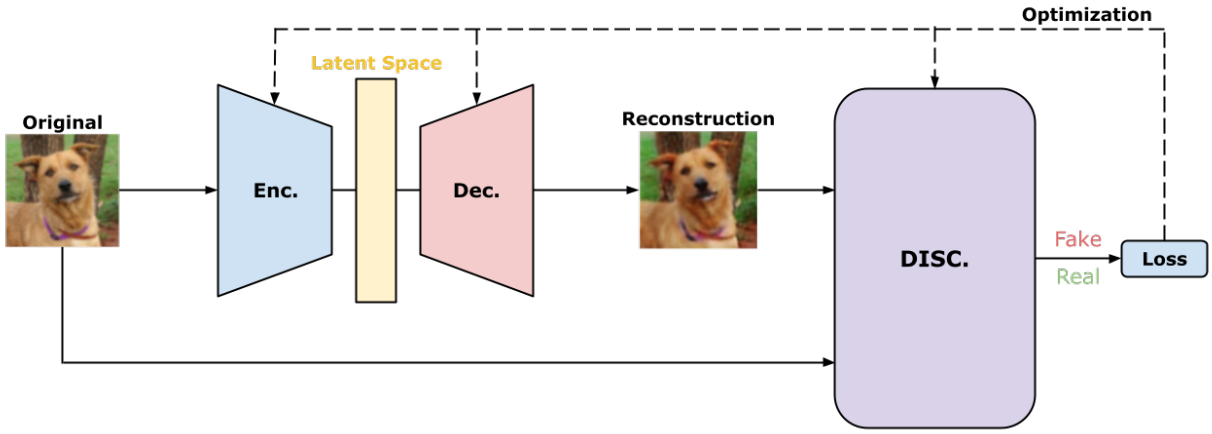

Figure 5: Diagram of the complete model architecture. The model is a CAE trained within an adversarial framework using a PatchGAN discriminator. The complete system consists of three main components: an encoder (Enc.), a decoder (Dec.), and a discriminator (Disc). The encoder and decoder together form the generator, which aims to form the reconstructed image. The discriminator promotes perceptual quality by distinguishing real from fake reconstructions at the patch level. The diagram highlights how input images are compressed, reconstructed, and evaluated across stages using adversarial and perceptual feedback.

Algorithm 1 Forward and Backward Passes through the Model Training

Input: Image batch $x \sim \mathcal{X}$; Networks E, G, D
Params: θ_E, θ_G, θ_D
for each step t **do**
  Sample real images $x \sim \mathcal{X}$
  $z \leftarrow E(x)$; $\hat{x} \leftarrow G(z)$
  $\mathcal{L}_{rec} \leftarrow \|x - \hat{x}\|_1 + \lambda_{LPIPS} \cdot \text{LPIPS}(\hat{x}, x)$
  if $t <$ `START_DISC` **or** $t \bmod 2 = 0$ **then**
    if $t \geq$ `START_DISC` **then**
      $\mathcal{L}_{GAN} \leftarrow -\mathbb{E}[D(\hat{x})]$
      $w \leftarrow \frac{\|\nabla_{\theta_G} \mathcal{L}_{rec}\|_2}{\max(\|\nabla_{\theta_G} \mathcal{L}_{GAN}\|_2, \epsilon)}$
      $\mathcal{L}_G \leftarrow \mathcal{L}_{rec} + \lambda_{GAN} \cdot w \cdot \mathcal{L}_{GAN}$
    else
      $\mathcal{L}_G \leftarrow \mathcal{L}_{rec}$
    end if
    Update θ_E, θ_G using $\mathcal{L}_G$
  end if
  if $t \geq$ `START_DISC` **and** $t \bmod 2 = 1$ **then**
    $\mathcal{L}_D \leftarrow \mathbb{E}[\text{ReLU}(1 - D(x))] + \mathbb{E}[\text{ReLU}(1 + D(\hat{x}))]$
    Update θ_D using $\mathcal{L}_D$
  end if
  if $t \bmod 200 = 0$ **then**
    Visualize $\hat{x}$
  end if
end for

Loss Functions The total generator loss combines three components:

$$\mathcal{L}_G = \mathcal{L}_{L1} + \lambda_{LPIPS} \cdot \mathcal{L}_{LPIPS} + \lambda_{GAN} \cdot w \cdot \mathcal{L}_{GAN} \quad (5)$$

where w is an adaptive weight calculated from the ratio of gradient norms of perceptual and adversarial losses, and $\lambda_{LPIPS}, \lambda_{GAN}$ are hyperparameters. These hyperparameters were tuned using adaptive weight w to dynamically balance perceptual and adversarial losses during training. To ensure training stability, w was clamped in a reasonable range (0 to 10,000).

4.3 Performance Metrics

To quantitatively assess the proposed model, this paper evaluates image compression and reconstruction quality using the following metrics: CR, PSNR, and SSIM. These metrics provide an understanding of both compression efficiency and the perceptual fidelity of the reconstructed images.

CR CR measures the reduction in data size achieved by the encoder. It is defined as the ratio of the original image size to the compressed image size:

$$\text{CR} = \frac{\text{Original Image Size (bytes)}}{\text{Compressed Image Size (bytes)}} \tag{6}$$

A higher CR indicates a greater reduction in data size. However, excessive compression can lead to degradation in perceptual quality, so it is critical to balance compression with the reconstruction output.

PSNR PSNR is a widely used metric that quantifies reconstruction accuracy by comparing the original and reconstructed images. It is calculated using MSE and is expressed in decibels (dB):

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \tag{7}$$

where MAX is the maximum possible pixel intensity value (e.g., 255 for 8-bit images). Higher PSNR values indicate that the reconstructed image is closer to the original in pixel-wise similarity. PSNR may not fully capture perceptual quality.

SSIM SSIM is a perceptual metric that evaluates the visual similarity between the original and reconstructed images by considering luminance, contrast, and structural information. Unlike PSNR, SSIM better reflects human visual perception:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{8}$$

where μ_x and μ_y are the local means, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the covariance of the original and reconstructed images x and y, and C_1 and C_2 are small constants to stabilize the division. SSIM values range from 0 to 1, with higher values indicating greater structural similarity.

Together, CR, PSNR, and SSIM provide a balanced evaluation framework: CR captures compression efficiency, PSNR assesses numerical fidelity, and SSIM gauges perceptual and structural integrity.

5 Results and Discussion

This section presents an analysis of the proposed compression framework through a series of ablation experiments, performance metrics, and qualitative visual output. An additional exploration and analysis of the individual model section serves to present their effects towards image compression. The progression from basic CAEs to a full model with adaptive loss weighting and perceptual enhancements is shown to improve visual quality and reconstruction fidelity.

5.1 Output Examples

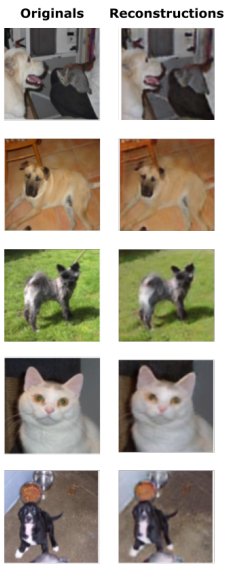

Figure 6: Qualitative output of the full model. Each image shows the result of compression and reconstruction through the model architecture. The reconstructions retain global structure and local detail, demonstrating effective balance between compression and fidelity.

The complete model, a CAE with ResBlocks, PatchGAN, and perceptual loss functions, aims to compress the data while maintaining high-fidelity reconstruction in both perceptual and pixel-level terms. Figure 6 shows the qualitative results of the model

through its reconstructions. The reconstructions of the complete model, although not perfectly replicating the original, still retain its perceptual similarity and key features. Therefore, the model yields the most visually faithful reconstructions and maintains a balance between perceptual sharpness and pixel-wise similarity.

5.2 Abalation Study

To understand the individual contributions of the architectures to the overall image compression performance, we also conducted a thorough ablation study. This includes both qualitative and quantitative evaluations of 3 successive model configurations.

- **CAE**
- **CAE + Residual Blocks [CAERB]**
- **CAE + Residual Blocks + PatchGAN + Perceptual Loss [GANCAERB]**

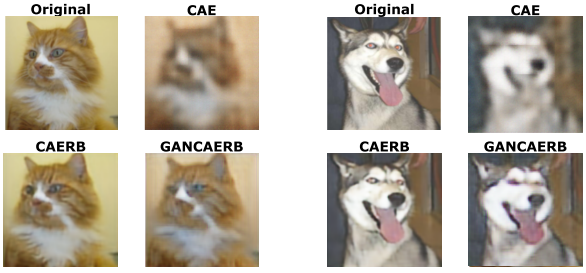

Figure 7: Ablation study showing reconstruction results for three configurations: CAE, CAERB, and GANCAERB. Visual differences show progressive improvement in detail preservation and realism. The CAE output generates blurry reconstructions with checkerboard artifacting, particularly in textured regions. The CAERB does improve edge retention, increases pixel similarity, and suppresses the artifacting. However the smaller details and textures of the image and dulled and blurred over. In the GAE-CAERB output, these problems are fixed.

Qualitative Comparison Figure 7 illustrates the reconstructed outputs of the 3 models in common input images. With the addition of a GAN in the GANCAERB model, both perceptual loss and adversarial loss are incorporated. However, when a GAN is added without proper loss balancing, the model can suffer mode collapse, which can distort the reconstructions, as shown in Figure 8. Once LPIPS, hinge loss, and adaptive loss weighting are incorporated, the final image result stabilizes in adversarial training, improving realism, and reducing edge artifacts.

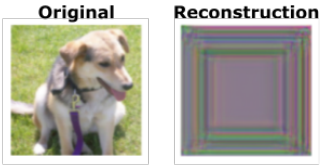

Figure 8: Reconstruction from a GAN-augmented CAE without perceptual or adaptive loss balancing. The output exhibits mode collapse and structural artifacts, demonstrating the importance of proper loss weighting.

Quantitative Performance Table 2 summarizes the trade-offs between CR and image quality between the models. CAE achieves the highest CR (48:1), at the cost of visual quality. CAERB and GANCAERB reduce compression efficiency (12:1) but improve PSNR and SSIM scores. Figures 9 and 10 display the PSNR and SSIM scores during training, indicating that the addition of ResBlocks, GANs, and perceptual losses results in further improvements in PSNR and SSIM. These models show a better balance between compression and reconstruction quality during training.

Table 2: Performance Metrics

Model	CR	PSNR (dB)	SSIM
CAE	48:1	23.00	0.6483
CAERB	12:1	28.86	0.8911
GANCAERB	12:1	22.95	0.7433

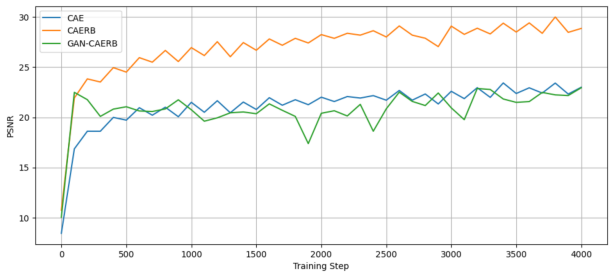

Figure 9: PSNR tracked during training of the three model. Shows gradual improvement in pixel-level similarity between input and reconstructed images over training steps.

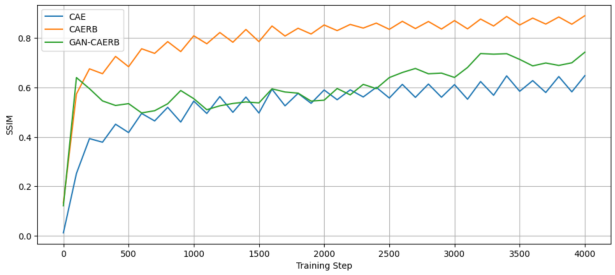

Figure 10: SSIM over training steps for the three models. Reflects improved structural and perceptual similarity between the input and reconstruction.

Summary The ablation study confirms that each architectural enhancement contributes meaningfully to the overall performance of the model. The CAE provides a compact latent representation, the CAERB enhances image clarity, and the GAN-CAERB integrates perceptual cues to improve detail and realism. Together, they produce a compact yet high-fidelity compression system.

5.3 Limitations & Future Work

The current model faces several limitations resulting from computational resources and architectural choices. Despite the small Cats and Dogs dataset, computational power and resources power hindered depth and stability in training. Additionally, the model's latent space lacks interpretability, limiting semantic control and downstream applications like editing in the latent space. The focus on dimensionality rather than byte-level optimization further reduces its effectiveness in bandwidth-constrained scenarios. To address these issues, future work will explore VQ-VAE architectures, which offer discrete latent representations, improved training stability, and better alignment with real-world storage constraints. Scaling up to larger, more diverse datasets will also enhance generalization and industrial viability.

6 Conclusion

This work presented a lightweight, end-to-end hybrid model for image compression, which combined CAE, ResBlock, and a GAN architecture to improve reconstruction quality. This experiment focused on analyzing the effects of these components using standard metrics such as CR, PSNR, and SSIM. The results demonstrate that a lightweight model can achieve competitive performance while remaining a more generalized solution. By making image compression more efficient and lightweight, we take a small but critical step toward reducing the data storage burdens associated with edge applications, making deep learning compression deployable for real-world tasks.

References

[1] E. AI and V. Alliance. Edge ai and vision 2023 report, 2023. https://www.edge-ai-vision.com/.

[2] Y. Bai, X. Yang, X. Liu, J. Jiang, Y. Wang, X. Ji, and W. Gao. Towards end-to-end image compression and analysis with transformers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 104–112, 2022.

[3] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[4] T. Boutell. Png (portable network

graphics) specification version 1.0, 1997. https://www.w3.org/TR/PNG/.

[5] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Deep convolutional autoencoder-based lossy image compression, 2018.

[6] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in neural information processing systems*, 28, 2015.

[7] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[8] Food and A. O. of the United Nations. Smart farming and data-driven agriculture: The fao perspective, 2021. https://www.fao.org/documents/card/en/c/CB6642EN/.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] P. Karanis. 2018-zhang et al-parasitology research. 03 2018.

[12] L. Li, Y. Fang, J. Wu, and J. Wang. Autoencoder based residual deep networks for robust regression prediction and spatiotemporal estimation. *arXiv preprint arXiv:1812.11262*, 2018.

[13] J. Ma, H. Liu, Y. Li, J. Wang, and L. Ding. Unet-compression: A unet-inspired neural architecture for learned image compression. *IEEE Transactions on Multimedia*, 2023.

[14] F. Mentzer, E. Agustsson, R. Timofte, and L. Van Gool. High-fidelity generative image compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] A. K. Naveen, S. Thunga, A. Murki, M. Kalale, and S. Anil. Autoencoded image compression for secure and fast transmission. In *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, pages 1–6. IEEE, 2024.

[16] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[17] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[18] G. K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

[19] W. Wang, Y. Huang, Y. Wang, and L. Wang. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 490–497, 2014.

[20] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network, 2017.

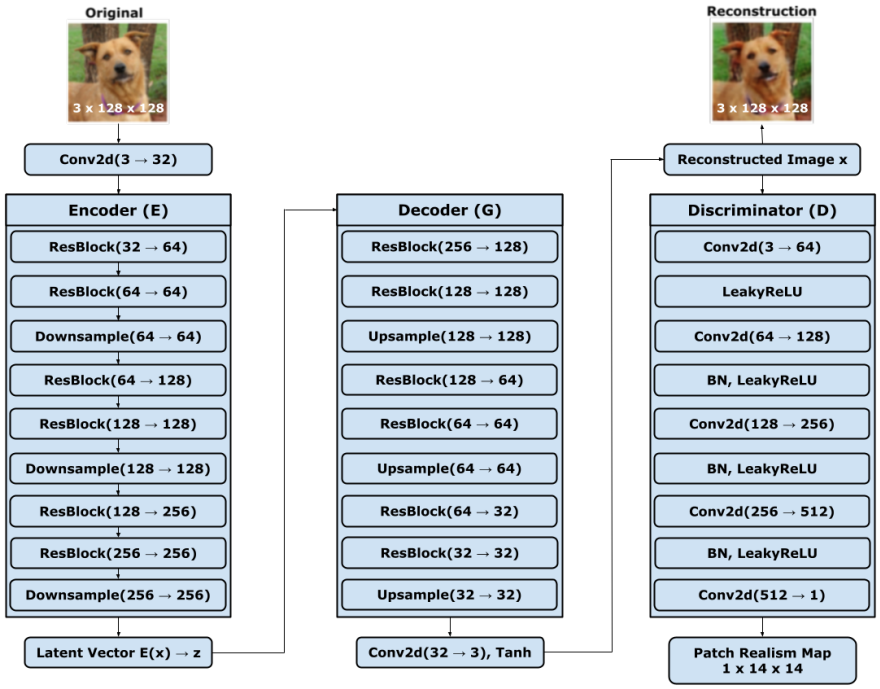

Figure 11: High-level block diagram of the proposed hybrid compression model. Displays how each component (CAE, ResBlocks, GAN, and perceptual loss functions) interacts within the end-to-end architecture. Each residual block consists of two convolutional layers with batch normalization and ReLU activations, and includes skip connections. Between groups of residual blocks, spatial resolution is reduced using strided convolutions. At the end of the decoder, a final 3×3 convolutional layer followed by a tanh activation produces the reconstructed image $\hat{x} \in \mathbb{R}^{3\times128\times128}$. Each layer uses a 4×4 kernel with a stride of 2 and is followed by LeakyReLU activations and batch normalization (except the first and last layers). The final output is a spatial map $D(x) \in \mathbb{R}^{1\times14\times14}$, representing the realism score of each patch in the input image.

Figure 12: Reconstructed output from the baseline CAE model. The image is visibly blurry with noticeable artifacts due to reliance on pixel-wise loss and absence of perceptual or adversarial components.

Figure 13: Reconstruction from CAERB. Enhanced edge retention and fewer checkerboard artifacts demonstrate improved feature representation and stability during training.

Figure 14: Reconstruction using LPIPS perceptual loss and adaptive loss weighting. The image preserves both high-frequency details and global structure, reflecting the benefits of perceptual optimization. Reconstruction result with full model including hinge loss. The image exhibits sharp edges, balanced texture, and minimal adversarial artifacts, indicating stabilized GAN training.