Neural Compression System

A Comprehensive Study on Codebook Efficiency for MNIST Dataset Compression

Yann Guszkiewicz with the assistance of certain LLM

Abstract

This work investigates neural image compression using Vector Quantized Variational Autoencoders (VQ-VAE), focusing on how architectural design enables effective tokenization of visual data. By converting continuous image features into discrete latent tokens, VQ-VAE provides a powerful mechanism for compressing image datasets that exhibit structural recurrence. Using the MNIST handwritten digit dataset as a testbed, I explore how neural networks can learn compact, semantically meaningful representations suitable for storage and reconstruction. The architecture, based on convolutional encoders and a learned codebook, enables up to an 18:1 compression ratio while preserving high visual fidelity. Rather than aiming to minimize distortion through classic rate-distortion trade-offs, this study emphasizes the capacity of neural models to exploit data redundancy via learned patterns. The results suggest that such approaches are best suited for structured datasets, where internal regularities can be leveraged for efficient compression.

1 Introduction

As the volume of visual data continues to grow exponentially, traditional compression techniques—whether lossless or handcrafted lossy methods—struggle to keep pace with the need for efficient storage and transmission. Neural compression offers an alternative by learning data-specific representations that compress not only pixels, but also the statistical and structural properties of images. Vector Quantized Variational Autoencoders (VQ-VAE) represent a key approach in this domain. By encoding inputs into discrete latent variables through vector quantization, they enable token-based representations that are both compact and semantically expressive. These tokens can be stored, transmitted, and decoded by a shared model, yielding reconstructions that maintain high perceptual quality. In this work, I explore the effectiveness of VQ-VAE-based compression on the MNIST dataset. While MNIST is visually simple, it exhibits strong internal regularities—making it an ideal setting for analyzing how neural architectures can identify and exploit recurring patterns. I demonstrate how architectural choices—such as latent dimensionality and quantization strategy—affect both compression performance and reconstruction quality. Rather than treating codebook size as a tuning parameter, this study uses it as a lens to understand the broader relationship between model structure, pattern learning, and compressibility.

2 Related Work

2.1 Variational Autoencoders

VAEs, introduced by Kingma and Welling, model continuous latent spaces for data generation. However, their continuous nature poses limitations in tasks requiring discrete representations.

2.2 Vector Quantization and VQ-VAE

Vector quantization transforms continuous feature spaces into discrete indices. VQ-VAE integrates this into the autoencoder structure via a learned codebook and a straight-through estimator, enabling end-to-end differentiability. This approach is foundational for token-based representation learning in vision.

2.3 Neural Compression

Recent works (e.g., Ballé et al., Mentzer et al.) have pushed neural image compression beyond classical codecs. However, most emphasize rate-distortion optimization. Here, the focus shifts to architectural choices, particularly the codebook, and their effects on compression and reconstruction.

3 Methodology

3.1 Architecture Overview

The architecture consists of a convolutional encoder, a vector quantizer, and a convolutional decoder. The encoder uses two convolutional layers with 4×4 kernels, stride-2, transforming input channels from 1 to 128 to 64. Each 28×28 MNIST image is compressed into a $7 \times 7 \times 64$ latent representation, representing a 16-fold spatial compression. The decoder mirrors this structure using transposed convolutions with ReLU and Sigmoid activations. Quantized latents are then reconstructed through the decoder.

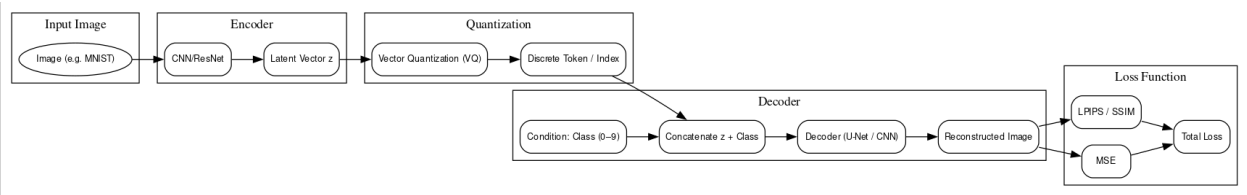

Figure 1: VQ-VAE Architecture Pipeline

3.2 Tokenization and Encoding

Encoded latent vectors are quantized using the nearest codebook embedding via L2 distance minimization. The vector quantizer maintains embeddings of dimension 64, initialized uniformly in $[-1/K, 1/K]$ where K is the codebook size. The quantized indices serve as compressed representations. Each token is an integer in $[0, K)$, where K is the codebook size. These tokens are serialized into compact data structures for storage or downstream use.

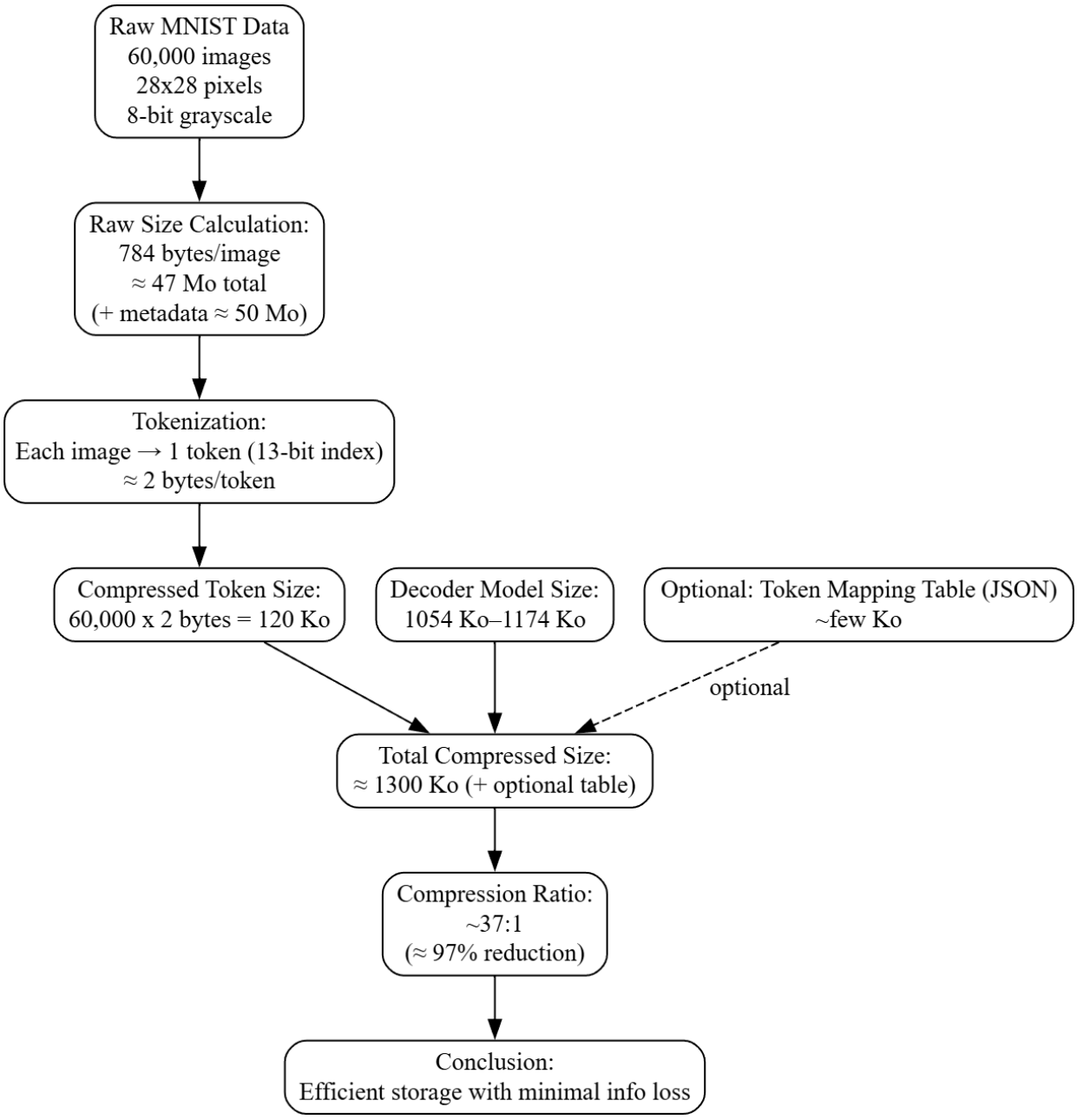

Figure 2: Compression workflow: input images are encoded into discrete tokens via vector quantization, stored as integer arrays, and later reconstructed via the decoder.

3.3 Training Protocol

The loss function is a combination of reconstruction loss and vector quantization loss:

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \mathcal{L}_{VQ}$$

where $\mathcal{L}_{VQ}$ includes both embedding loss and commitment loss with $\beta = 0.25$.

Training is conducted using Adam optimizer (learning rate: 2×10^{-4}), batch size of 64, and 3 epochs. Due to the relatively simple nature of MNIST digits and the goal of observing the model's capacity to learn the training data extensively, this reduced training schedule is intentionally chosen to allow for potential overfitting. The commitment cost $\beta = 0.25$ is used to stabilize encoder–quantizer interaction.

4 Experiments and Results

4.1 Metrics

Evaluation is based on Mean Squared Error (MSE), Structural Similarity Index (SSIM), and the total model size in kilobytes (Ko). All experiments were run using consistent training and inference pipelines. For fair comparison, metrics are computed on the first 100 test images to ensure statistical consistency across codebook sizes.

4.2 Quantitative Results

Codebook Size	MSE	SSIM	Model Size (Ko)
64	0.0188	0.6901	1062
128	0.0094	0.8713	1078
256	0.0126	0.8212	1110
512	0.0123	0.8597	1174

Table 1: Comparison of reconstruction quality and model size for different codebook sizes.

4.3 Analysis

As seen in Table 1, the model with 128 codebook entries outperforms others in both MSE and SSIM. The model size increases gradually with codebook size due to embedding table expansion, but reconstruction quality does not follow the same trend.

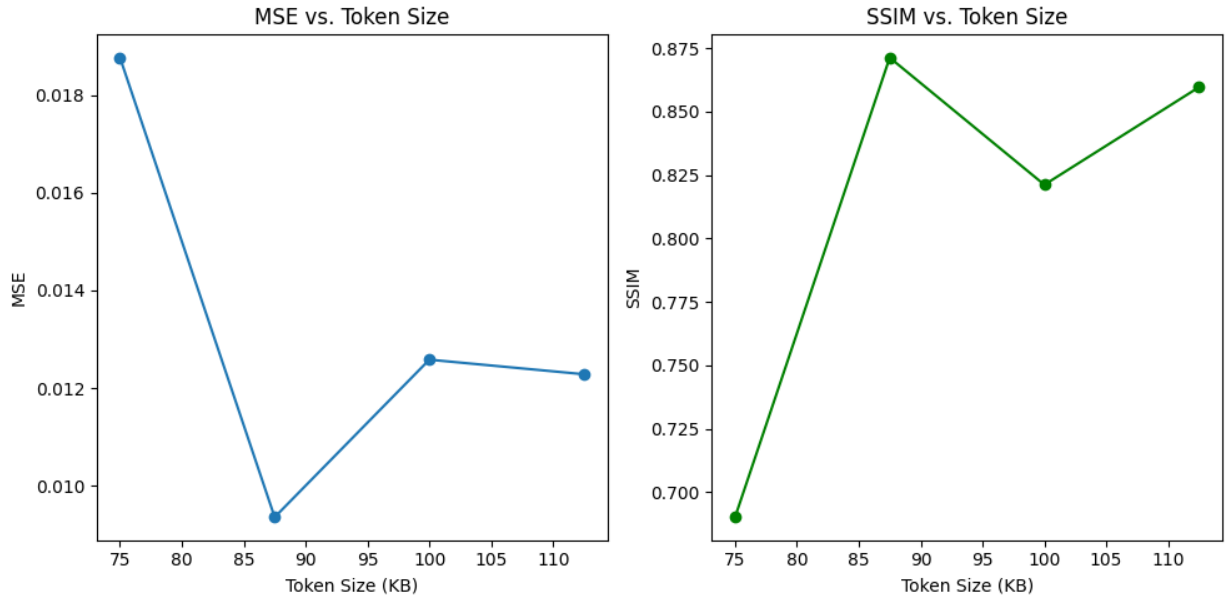

Figure 3: MSE and SSIM vs. Codebook Size

4.4 Visual Reconstructions

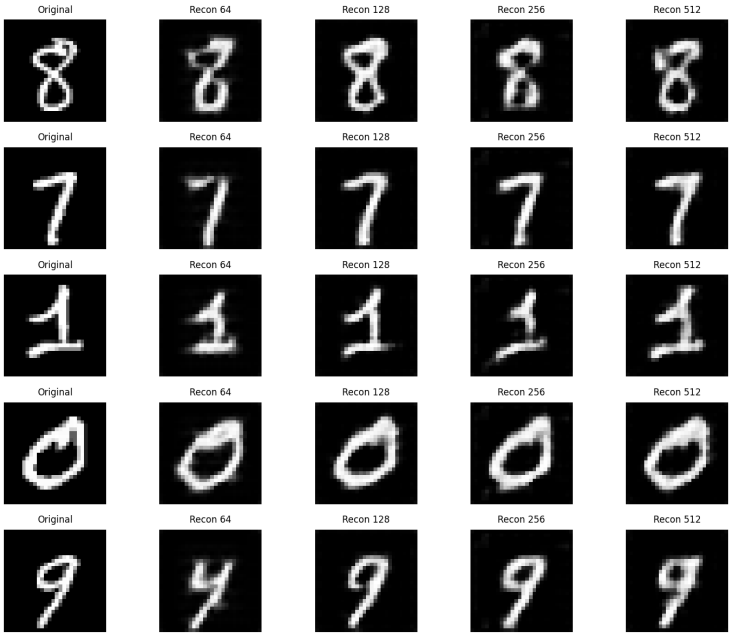

Figure 4: Reconstructed digits for different codebook sizes

4.5 Discussion

Contrary to expectation, increasing the codebook size beyond a certain point degrades performance slightly. This is not necessarily a failure of overfitting, but rather an inefficiency stemming from the mismatch between the codebook's capacity and the inherent complexity of the MNIST dataset. Larger codebooks have the potential to learn and represent more intricate patterns. However, since MNIST digits are relatively simple and lack highly complex features, these larger codebooks may attempt to capture nuances that are non-existent or mere noise, effectively underutilizing their increased capacity for meaningful representation. The 128-vector codebook provides the best overall balance, capturing essential structural features without introducing unnecessary complexity or redundancy.

5 Conclusion

This study explores the use of VQ-VAE neural architectures for image compression, emphasizing how their structure enables efficient token-based representations of visual data. The results show that meaningful compression ratios can be achieved approximately 18:1 for MNIST by discretizing images into a fixed number of tokens using a learned codebook. The most effective setup encodes each image into 49 tokens using a codebook of 128 vectors, yielding high reconstruction quality with minimal storage cost. However, the contribution of this work lies less in selecting an optimal codebook size and more in validating that such architectures can exploit recurring patterns for compression. Because this method depends on the ability to recognize and reuse structural features, it is particularly effective on datasets like MNIST that exhibit strong internal consistency. Despite promising results, it is important to acknowledge that this approach has limitations. The overhead of storing the neural model (approximately 1 MB) must be amortized across large datasets to be competitive with traditional codecs. Moreover, since this architecture relies on learning pattern recurrence, it may struggle on datasets with high visual entropy or little structural regularity. Future directions include extending this architecture to adapt to more diverse datasets and investigating whether dynamically scaling the codebook or using hierarchical tokenization could further improve generalization and compression quality in complex image domains.