Neural Compression System is All You Need

Classification and Semantics from Discrete Latent Codes

Yann Guszkiewicz with the assistance of certain LLM

Abstract

This follow-up study demonstrates that discrete token representations obtained from a pretrained Vector Quantized Variational Autoencoder (VQ-VAE) are sufficient not only for reconstructing images but also for downstream tasks like classification. By directly feeding the token indices into a lightweight classifier, we evaluate the semantic richness of these compressed representations. Our results show that classification accuracy remains high (94.8%) even when using token sequences instead of raw pixels, highlighting the power of neural compression as a feature extraction mechanism. Moreover, the compressed classifier model is 5.8× smaller and 15× faster to train compared to a traditional convolutional neural network (CNN) while maintaining competitive performance. This work positions tokenized representations as a minimal, unified interface for both storage and inference.

1 Introduction

Modern neural compression systems, such as VQ-VAE, enable transformation of images into compact sequences of discrete codes. While prior work has emphasized reconstruction fidelity and compression ratios, this study explores a different axis: *can we treat these tokens as sufficient input for downstream machine learning tasks?*

If neural compression tokens retain task-relevant semantics, then they should suffice as inputs to a classifier. This would mean the encoder has implicitly learned to disentangle features that support both compression and discrimination. We evaluate this hypothesis by training a simple token-based classifier on top of VQ-VAE tokens and comparing it with a baseline CNN trained directly on images.

2 Methodology

2.1 System Overview

The system consists of two stages:

1. A pretrained VQ-VAE encoder compresses MNIST images into 49 discrete tokens (indices in a 128-codebook).
2. A classifier is trained using only these 49-token sequences as input.

Figure 1: System architecture: images are encoded into token sequences via a frozen VQ-VAE encoder; tokens are classified via a lightweight MLP.

2.2 Classifier Architecture

The token classifier embeds each token (integer in $[0, 127]$) into a 32-dimensional vector and flattens the sequence:

$$x \in Z^{49} \rightarrow \text{Embedding} \rightarrow R^{49\times 32} \rightarrow \text{MLP}$$

The MLP consists of two linear layers:

$$\text{Linear}(1568, 128) \rightarrow \text{ReLU} \rightarrow \text{Linear}(128, 10)$$

3 Experiments

3.1 Setup

We compare the performance of two models:

- **Baseline CNN:** trained directly on 28×28 grayscale MNIST images.
- **Token Classifier:** trained on 7×7 token sequences generated by a frozen VQ-VAE encoder.

All experiments use the same training data and number of epochs. Metrics include accuracy, model size, number of parameters, and training time.

3.2 Results

Model	Accuracy	Train Time (s)	Params	Size (KB)	Input
CNN Baseline	98.67%	128.5	1.20M	4690.7	Raw Pixels
Token Classifier	94.76%	8.6	206k	808.4	Token IDs

Table 1: Comparison of CNN and Token Classifier on MNIST classification

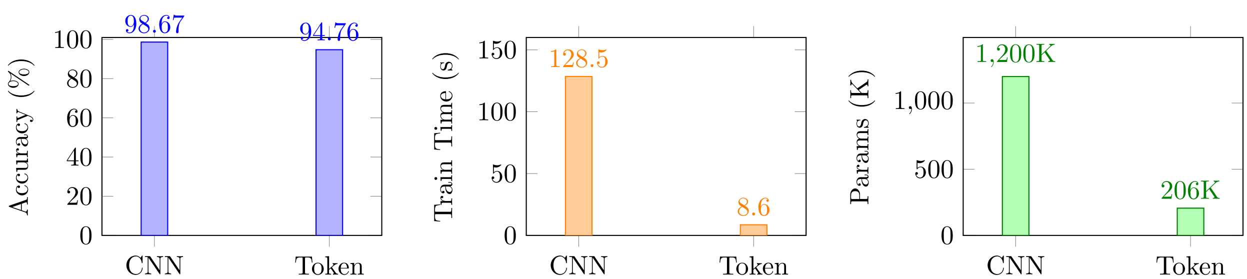

Figure 2: Visual comparison of metrics between CNN and Token Classifier

3.3 Interpretation

The token classifier achieves 94.76% accuracy while being nearly $6\times$ smaller and $15\times$ faster to train. Despite operating on compressed, discretized representations, it retains most of the classification power of a full CNN.

This suggests that the encoder learns highly structured, semantically relevant features, and that vector quantization preserves task-critical information. In essence, the VQ-VAE encoder compresses the input into a task-agnostic form that can be used for both reconstruction and recognition.

4 Discussion

The results imply that neural compression is more than a storage trick — it's a semantic bottleneck. By training a decoder and classifier on the same tokens, we demonstrate that these embeddings support multi-purpose computation. The encoder acts as a universal front-end for both reconstruction and classification.

This property is critical for edge and embedded systems, where storage, bandwidth, and computation are constrained. A device could encode images once and send only tokens, which could then be decoded or classified on the cloud, depending on the task.

5 Conclusion

We have shown that compressed latent codes generated by a VQ-VAE contain sufficient information for classification. This elevates neural compression to a dual-purpose mechanism: reducing redundancy while preserving semantics.

Key Takeaways

This study demonstrates that tokens produced by neural compression pipelines can be effectively reused for tasks beyond image reconstruction. A lightweight classifier operating directly on token indices achieves nearly 95% accuracy on MNIST, underscoring the semantic quality of the latent representations. This approach opens the door to task-agnostic pipelines, where compression acts not only as a memory-efficient format but also as a practical interface for inference.