Quantized Convolutional Autoencoder for Video Compression

Gaurav Mitra
Dept. of Aerospace Engineering
University of Texas at Austin
Austin, United States
mitragaurav10@gmail.com

***Abstract*— Video compression is a crucial task for efficient storage and transmission of video data. This paper presents a convolutional neural network (CNN) autoencoder designed for lossy video compression using a quantized latent representation. The proposed model encodes sequences of video frames into a discrete latent code space, enabling a tunable rate-distortion trade-off. The model is trained and evaluated on a dataset of grayscale video sequences (Moving MNIST) to assess reconstruction quality at various compression levels. Experimental results demonstrate that the autoencoder successfully compresses and reconstructs video frames, and the rate-distortion analysis shows expected trade-offs between bitrate (quantization codebook size) and reconstruction error. The model architecture and training procedure are described in detail, and example reconstructions and a rate-distortion curve are provided. This work highlights the potential of learned autoencoders for video compression and discusses challenges such as training convergence for large codebooks.**

Keywords*— *Video compression, autoencoder, convolutional neural network, quantization, rate-distortion, deep learning

I. Introduction

Video streaming and storage demand effective compression techniques to reduce bandwidth and memory requirements. Traditional video codecs (e.g., H.264/AVC, H.265) rely on hand-crafted transform coding and motion prediction algorithms. In recent years, deep learning approaches have emerged as a promising alternative for image and video compression. In particular, autoencoder architectures have been applied to learn compact latent representations of visual data. For example, Habibian et al. propose a 3D convolutional autoencoder with a discrete latent space for video compression [1]. Similarly, variational autoencoders (VAEs) [2] introduced the idea of learning encodings that balance reconstruction fidelity and compression by optimizing a rate-distortion objective. These learned approaches can adapt to data characteristics and have shown competitive performance with classical codecs [1].

In this paper, a quantized convolutional autoencoder is developed for compressing video sequences. The autoencoder consists of an encoder network that transforms an input video (a sequence of frames) into a lower-dimensional latent representation, and a decoder network that reconstructs the video frames from this latent code. To achieve compression, quantization is imposed on the latent space: the continuous latent features are mapped to the nearest entries in a fixed-size codebook. This discretization limits the information (approximately corresponding to a fixed number of bits per latent element) and thereby controls the compression rate. By adjusting the codebook size (number of quantization levels), it is possible to navigate the trade-off between bitrate and distortion in the reconstructed video.

This approach is inspired by prior work on learned video compression [1] but uses a simplified architecture and training pipeline. The model is trained on a benchmark dataset of synthetic video clips (the Moving MNIST dataset, containing 10,000 sequences of moving handwritten digits) to evaluate its performance. Qualitative examples of compressed video reconstructions and quantitative analysis of rate-distortion behavior are presented. The results demonstrate that larger latent codebooks (higher bitrates) yield lower distortion, though with diminishing returns. Practical challenges such as training stability with large codebooks are also discussed.

The remainder of this paper is organized as follows: Section II describes the model architecture, quantization method, and training procedure. Section III presents experimental results, including visual examples and rate-distortion analysis. Section IV provides discussion of the findings and outlines directions for future work. Section V concludes the paper. An acknowledgment and references are provided at the end.

II. METHODOLOGY

A. Autoencoder Architecture

The proposed compression model is a convolutional autoencoder that operates on short video sequences. The autoencoder takes a sequence of grayscale frames as input and produces a reconstructed sequence as output. The encoder is a deep CNN that gradually reduces the spatial resolution of the frames while increasing the number of feature channels, thereby condensing the information. Specifically, the encoder uses 3D convolutional layers (convolutions across both spatial dimensions and time) with stride (1,2,2) and kernel size 5 in the first and last layers of the encoder stack. This configuration downsamples the height and width of feature maps by a factor of 2 at each layer (while preserving temporal length), effectively compressing the spatial detail. After several convolutional layers, a bottleneck latent representation is obtained. To improve representational capacity, a residual block is included in the bottleneck: inside this ResBlock, convolutional layers with $3 \ x \ 3$ kernels (stride 1) and 128 feature channels are used, along with skip connections. The decoder mirrors the encoder structure using transposed convolution (deconvolution) layers to upsample the features back to the original spatial resolution. The transpose conv layers use kernel size 5 and stride (1,2,2), inversely restoring height and width at each stage. Intermediate batch normalization layers are applied throughout to stabilize training. The overall architecture is summarized in Figure 1, which also illustrates the encoder-decoder structure of the network.

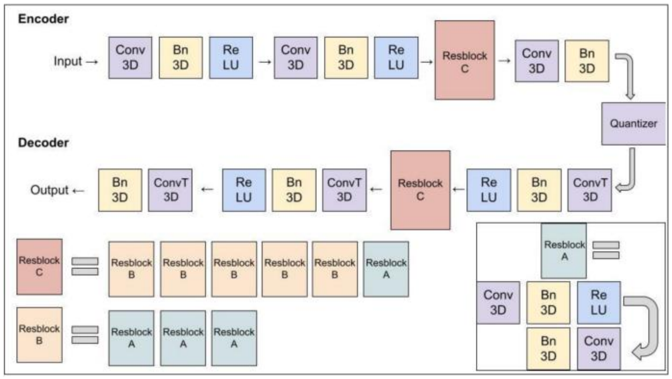


Fig. 1. Convolutional autoencoder architecture for video compression. The encoder compresses the input video frames into a latent code, and the decoder reconstructs the frames from this code. Residual connections within the bottleneck improve learning, and quantization is applied on the latent features (not shown in this diagram).

B. *Latent Quantization*

A key feature of this model is the quantized latent space, which enables a controllable compression rate. Instead of allowing the encoder to produce arbitrary continuous values, the latent representation is restricted to a discrete set of values. A codebook of fixed size K is implemented. After the encoder transforms the input into a latent feature map, each element of this latent map is quantized to the nearest entry in the codebook. This process assigns each latent vector an index from 1 to K. Because quantization is non-differentiable, straight-through estimation is used in training to allow gradients to flow as if the operation were identity in the backward pass. The codebook entries can be learned or fixed; in this implementation they are initialized uniformly and slightly adjusted through gradient updates.

By varying the codebook size K, the bitrate is directly influenced. A larger codebook can represent the latent space with more precision (higher rate, since each latent element requires $log_2 K$ bits), potentially reducing distortion. Conversely, a small codebook (e.g., $K = 2$) greatly compresses the data at the cost of higher distortion. This forms the basis of the rate-distortion analysis in the experiments.

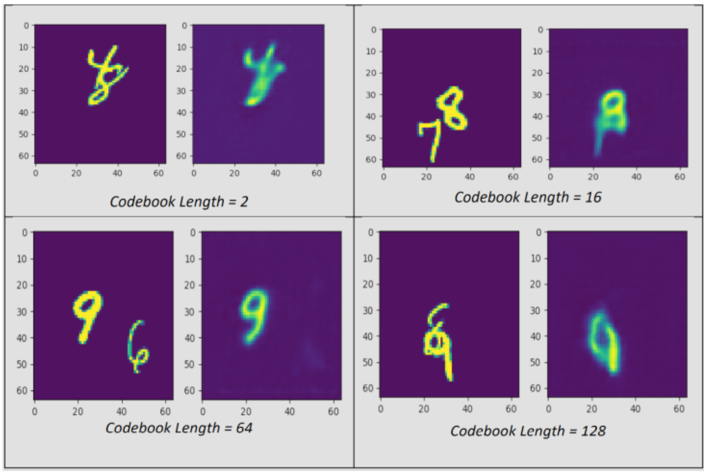
C. *Training Procedure*

The quantized autoencoder was trained on the Moving MNIST dataset, which contains short 64×64 grayscale video sequences of moving handwritten digits. Each training sample is a sequence of 20 frames, input as a 3D tensor (time × height × width, one channel). The model was implemented in PyTorch and optimized using Adam with mean squared error (MSE) loss between reconstructed and original frames.

Training began with a large codebook ($K = 128$) to establish reconstruction ability. Subsequently, models were trained for $K \in \{2, 4, 8, 16, 32, 64, 128\}$ under identical hyperparameters (learning rate, batch size, epochs). Early stopping was applied if validation-loss reduction fell below 1 %. Although intended to detect convergence, this criterion occasionally halted training prematurely for large K. Throughout training, reconstruction loss and sample outputs were monitored to ensure effective learning. Each model yielded a point on the rate-distortion curve corresponding to its codebook size and reconstruction error.

III. EXPERIMENTAL RESULTS

After training with various quantization levels, reconstruction quality was evaluated qualitatively and quantitatively. Figure 2 shows example reconstructions for $K = 2, 16, 64, 128$. As expected, a very small codebook ($K = 2$) produces blurry reconstructions, while larger codebooks yield higher-fidelity results closer to the original frames.



Codebook Length = 2

Codebook Length = 16

Codebook Length = 64

Codebook Length = 128

Fig. 2. Example reconstructed video frames at different compression levels.

To quantify compression performance, a rate-distortion curve is plotted (Figure 3), with effective bitrate (bits per pixel) on the x-axis and reconstruction distortion (MSE) on the y-axis. Distortion decreases as codebook size grows from 2 to 128, with diminishing returns beyond $K = 64$. Models with $K = 64$ and 128 did not fully converge before early stopping, leaving some performance potential unrealized, but the overall trend validates the expected trade-off.

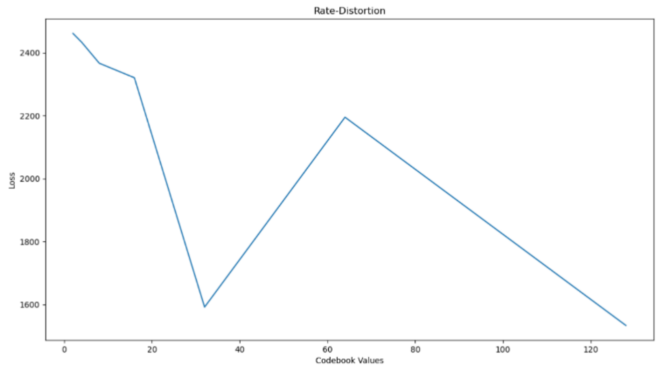


Fig. 3. Rate-Distortion curve for the quantized autoencoder.

IV. DISCUSSION

The results demonstrate that the convolutional autoencoder learns to compress and reconstruct video sequences effectively, with performance scaling as latent quantization becomes more fine-grained. This approach integrates a learned neural compression scheme with explicit bitrate control through the codebook size, consistent with the concept of rate-distortion autoencoders [1].

Training with large codebooks ($K = 64, 128$) was challenging because the relative-improvement threshold stopped optimization early. Allowing more epochs or adopting a different convergence criterion would improve high-capacity performance. Future work could also train a single model to operate at multiple rates through conditional codebook usage.

Additionally, the current autoencoder optimizes only reconstruction error and does not explicitly minimize entropy. A complete compression system would optimize a weighted sum of distortion and bitrate (entropy of the latent code) [1][2]. Learning an entropy model and incorporating it into the loss function could yield better bit-per-pixel efficiency.

The Moving MNIST results show that the autoencoder handles simple, low-resolution video data. For more complex content, the architecture could be scaled up and augmented with motion compensation or perceptual loss functions. The method is applicable to domain-specific compression tasks such as autonomous driving or remote sensing.

In summary, the quantized autoencoder approach is viable for video compression and offers a learnable alternative to hand-crafted codecs. The main challenges are training for large codebooks and integrating explicit bitrate control into the objective. With improved training and entropy coding integration, learned video compression models could outperform traditional codecs in specialized settings.

V. CONCLUSION

A quantized convolutional autoencoder for video compression has been presented and demonstrated on grayscale video sequences. The model learns to compress video frames into a discrete latent representation, enabling tunable compression rates by adjusting the latent codebook size. Experiments show a clear rate-distortion trade-off: increasing quantization levels improves reconstruction quality. Even with a simple architecture, the autoencoder produces reasonable reconstructions and captures expected trends, illustrating the promise of data-driven compression methods. This work provides a foundation for further research into learned video compression, including better training for high-capacity models and explicit rate control in the loss function. Future evaluation on more complex content and comparison with conventional codecs are planned.

ACKNOWLEDGMENT

The author would like to thank Dr. Takashi Tanaka and Ronnie Ogden for their mentorship and guidance throughout this project. Additionally, the author would like to thank Pranad Muttoju for his contributions to the project. This work was conducted as part of a research initiative at the University of Texas at Austin.

REFERENCES

[1] A. Habibian, T. van Rozendaal, J. M. Tomczak, and T. S. Cohen, “Video Compression with Rate–Distortion Autoencoders,” in Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV), Seoul, Korea, Oct. 2019, pp. 7032–7041.

[2] D. P. Kingma and M. Welling, “Auto–Encoding Variational Bayes,” in Proc. 2nd Int. Conf. on Learning Representations (ICLR), Banff, Canada, Apr. 2014. [Online]. Available: arXiv:1312.6114*)*