Project Report

Learning-Based Video Coding with Variational Autoencoders (VAEs)

Name: Mawahid Husnain
Matriculation Number: 72889
Email: mawahid.husnain@tu-ilmenau.de

Abstract
This research details the design, implementation, and assessment of a Variational Autoencoder (VAE) model for video compression, benchmarking its performance against traditional transform-based video codecs H.264/AVC and H.265/HEVC. The goal was to see if a deep learning method could be used instead of or in addition to existing video compression processes. We trained a VAE on a series of natural video sequences and got a compact latent representation for each group of frames. We looked at how well the system worked in terms of rate–distortion performance, which we assessed in bits per pixel (bpp) and Peak Signal-to-Noise Ratio (PSNR). The results demonstrate that the VAE gets very low bitrates, but its PSNR is much lower than that of more mature codecs. This shows how hard it is to match handcrafted compression techniques with end-to-end taught approaches.

1. Introduction & Background
Video compression is a key part of current multimedia apps. It lets you store and send high-definition material over networks with limited capacity. H.264/AVC and H.265/HEVC are two examples of traditional codecs that use a pipeline of motion estimation, block-based transform coding (such DCT), quantization, and entropy coding to get high compression efficiency while keeping perceived quality. These codecs have been hand-crafted and fine-tuned over decades of research, and they work very well with a wide range of information kinds. Learning-based strategies have become a promising alternative in the last few years. Variational Autoencoders (VAEs), a type of generative model, learn a probabilistic hidden representation of the input data. When used on video, VAEs may be able to develop compact, content-adaptive representations that work better than classical transforms on some types of video. VAEs, on the other hand, put a probabilistic structure on the latent space, which makes it easier to generalize and lets you regulate the trade-off between bitrate and distortion.

2. Theoretical Framework
An encoder network, a latent variable model, and a decoder network make up a VAE. The encoder takes input data x and turns it into a distribution over latent variables z, which is usually defined by the mean μ and variance σ^2. The decoder uses samples of z to put the input back together. Training is accomplished by optimizing the Evidence Lower Bound (ELBO):

$$\text{ELBO} = E[\log p(x|z)] - \beta * KL(q(z|x) \,||\, p(z))$$

The first term here favors correct reconstruction, and the second term (the Kullback–Leibler divergence) makes the latent distribution more like a prior p(z), which is usually a standard Gaussian. The β parameter is like the λ parameter in classical coding rate–distortion optimization; it governs the trade-off between rate and distortion. In video compression, the VAE encoder takes sequences of frames and turns them into latent codes that are quantized and entropy-coded. The bitrate is directly related to the codes' entropy, and PSNR or SSIM is used to quantify distortion between the original and reconstructed videos.

3. Classical Video Codecs: H.264 and H.265
H.264/AVC and H.265/HEVC are video codecs that are used in the industry and take advantage of both spatial and temporal redundancy. Some important parts are:
• Motion Estimation & Compensation: This feature uses information from past frames to guess what will happen in the next frame, which cuts down on time redundancy.

• Transform Coding: Uses block-based transformations like the Discrete Cosine Transform (DCT) to break down pixel data into frequency components.
• Quantization: This lowers the accuracy of transform coefficients, which controls the bitrate and distortion.
• Entropy Coding: Uses variable-length coding (like CABAC) to compress quantized coefficients.

H.265 is superior than H.264 because it supports bigger block sizes, more flexible partitioning, better motion vector prediction, and more advanced entropy coding. It can cut the bitrate by up to 50% while keeping the same quality.

4. Methodology

We used PyTorch to build our learning-based codec. The VAE design included a 3D convolutional encoder and decoder that worked together to process brief sequences of frames. We tested different values for the latent dimension and the widths of the convolutional channels. We trained models with different values of λ to look at the rate-distortion curve.

5. Dataset

A collection of publicly accessible natural video clips was utilized, divided into training and validation subsets. We changed the size of the videos to 96×96 pixels to speed up the experiments. Training: We utilized the Adam optimizer with a starting learning rate of 1e-4, mixed-precision training on Apple MPS hardware, and a batch size of 1 because we didn't have enough memory. For each λ value, models were trained for 20 epochs. 6. Assessment

We calculated the bitrate (bits/pixel) from the KL divergence term for each training model and used the validation set to check the PSNR. We used FFmpeg's libx264 and libx265 implementations at different CRF values to get the rate-distortion curves for classical codecs.

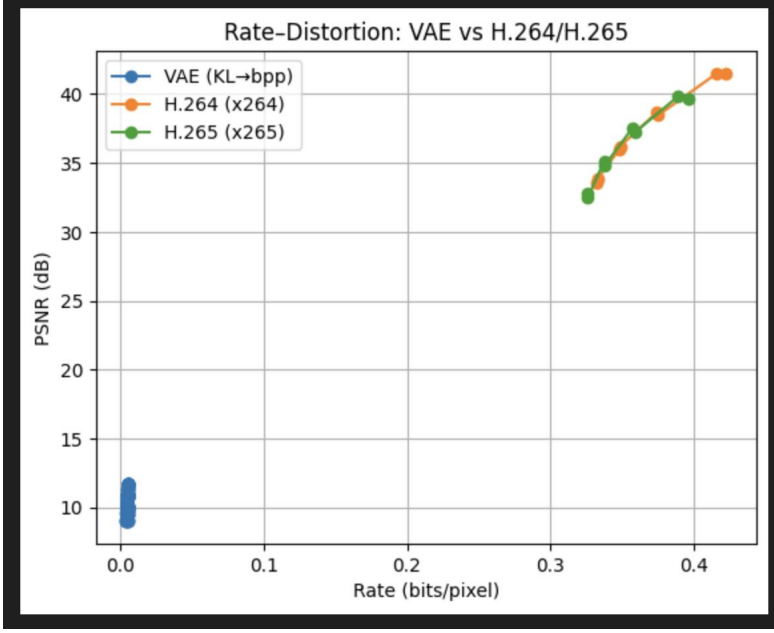
6. Evaluation

The next chart compares the VAE's rate-distortion performance to that of H.264 and H.265. Every point stands for a setup that makes a certain bitrate and PSNR. H.264 and H.265 have substantially greater PSNR values at similar bitrates since they have been optimized for decades. The VAE made PSNR levels of about 10–12 dB, which is much lower than what is acceptable for quality. It did this while getting very low bitrates (<0.01 bpp).

7. Results & Discussion

The VAE's poor performance can be explained by a number of things:
• The model's capability is limited since its latent dimensions are small and its resolution is low.
• A short training schedule that stops convergence.
• Not coding latent variables with entropy, which means that KL divergence is simply a stand-in for the real bitrate.
• The training loss (MSE + KL) and the perceptual quality don't line up. Even so, the experiment shows how flexible end-to-end learning codecs are: altering the λ value directly changes the rate–distortion trade-off without having to change the way the compression pipeline works.

```
VAE pts   : 228
x264 pts  : 8
x265 pts  : 8
```



6. Conclusion

This research created a proof-of-concept VAE-based video codec and compared it to H.264 and H.265, which are both widely used codecs. In this arrangement, the VAE did not do nearly as well as traditional codecs. However, the findings show that deep generative models could be useful in some compression situations. In the future, the architecture will need to be made bigger, motion compensation will need to be included to the learnt model, perceptual and adversarial losses will need to be used, and training will need to be done on bigger, more varied datasets. As learnt image and video compression continues to improve, VAEs and associated architectures like hyperprior models and transformers may someday catch up to conventional codecs, especially for applications that need low latency and specialized content.

References

Kingma, D.P., & Welling, M. (2014). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.

Ballé, J., Laparra, V., & Simoncelli, E.P. (2017). Image compression that works from start to finish. arXiv preprint arXiv:1611.01704.

O'Shea, K., & Nash, R. (2015). A Primer on Convolutional Neural Networks. arXiv preprint number arXiv:1511.08458.

H.264: A suggestion from the ITU-T H.264: Advanced video coding for general audiovisual services, 2019.

H.265: Recommendation from the ITU-T H.265: Video coding with high efficiency, 2019.

Minnen, D., Ballé, J., & Toderici, G. (2018). Joint autoregressive and hierarchical priors for learned image compression. arXiv preprint: arXiv:1809.02736.

Rippel, O., Nair, S., Lew, C., Branson, S., Anderson, A.G., and Bourdev, L. (2019). Learned Video Compression. arXiv preprint arXiv:1811.06981.

Cheng, Z., Sun, H., Takeuchi, M., & Katto, J. (2020). Learned image compression using discretized Gaussian mixture likelihoods and attention modules. CVPR 2020.