Compression Methods For Satellite Image Features

Christos Polzak {CLCP@stanford.edu}

Department of Electrical Engineering

Overview

- Satellite imagery of the entire world over time takes too much space, but many images are only seen by machine learning (ML) models
- Extracting image features and compressing the resulting vector drastically reduces space costs per image
- How do we optimally compress the feature vector to maximize compression and minimize the decrease in performance?

Satellite Image → (Frozen) Deep Feature Extractor (GeoSSL) [1] → PCA / Uniform Quantizer / PCA + Quantizer / Autoencoder / Entropy Bottleneck ❯ Linear Classifier / MLP Classifier

Main results:
- Uniform quantization is effectively lossless
- PCA + quantization is simple yet widely performs the best

Datasets

Training Dataset:
- Functional Map of the World [2] (abbreviated as fMoW)
  - high-resolution satellite imagery, 62 classes
  - labeled photos of buildings and structures from 200+ countries
  - normalized RGB images used
- examples: "flooded road", "zoo", "electricity substation", "barn"
- 360,000 train samples, 50,000 validation samples

"Barn"

"Space Facility"

Generalization Dataset:
- UCMerced Land Use dataset [3] (abbreviated as UCM)
  - high-resolution satellite imagery
  - 100 images each of 21 classes (e.g. "harbor", "forest", "runway")

Models and Methods

Compressors (and Decompressors)

Most-Variant Eigenvector Projection (PCA)

Estimate Covariance → Find K Largest Eigenvectors → Project to K largest (dim = K) → Reconstruct z

Uniform Quantization (Quant)

Clip outliers (up to ~.05%) → 2^N_bits uniform bins → Cast to N_bit type (dim = z_dim) → Approx. as bin medians

$$\mathrm{Int}\left(\frac{\max(z, z_{max} - \epsilon) - z_{min}}{z_{max} - z_{min}} \times 2^{\mathrm{N\_bits}}\right)$$ (where Int means truncate to integer)

Autoencoder

MLP [z_dim -> 1024 -> K_bottleneck] → Sigmoid (dim = K) → MLP [K_bottleneck -> 1024 -> z_dim] → MSE Loss

Entropy Bottleneck (EB) [4]

Conv Filters → Entropy Coder → Quantizer q(•) → Entropy Decoder → Rate-Distortion Loss

$$\mathcal{L}(z, \hat{z}) = -\log q(x) + \lambda \|z - \hat{z}\|_2^2$$ (where x = entropy-coded features)

Classifiers

Linear Classifier

$$x = Wz + b; \; y = \sigma_s(x), \quad \text{where } \sigma_s(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

$z \in \mathbb{R}^{2048}$; $W \in \mathbb{R}^{2048 \times c}$; $b, y \in \mathbb{R}^c$, where c is the number of classes

MLP Classifier

$$x_1 = \mathrm{Dropout}(\mathrm{ReLU}(W_1 z + b_1)) \text{ where } z \in \mathbb{R}^{2048}, \; x_1 \in \mathbb{R}^{2048}$$

$$x_2 = \mathrm{Dropout}(\mathrm{ReLU}(W_2 x_1 + b_2)) \text{ where } x_2 \in \mathbb{R}^{1024}$$

$$y = \sigma_s(W_3 x_2 + b_3)) \text{ where } y \in \mathbb{R}^c$$

Pipeline

fMoW Features (dim=2048) → Compressor Training Scheme

fMoW or UCM Features (dim=2048) → Compressor -> Decompressor → Decompressed Features (dim=2048) → Classifier

Key Results

Compressor Type	Data Size (MB)*	fMoW Linear Acc. (%)	fMoW MLP Acc. (%) (Δ_linear**)	UCM Linear Acc (%)
Baseline	3200	69.1	70.9 (+1.8)	94.5
Quant (N_bits=8)	818	***69.1***	***71.0 (+1.9)***	***94.5***
EB	124	67.0	68.1 (+1.1)	93.6
PCA (K=200)	338	66.8	*70.5 (+3.7)*	93.6
PCA (K=200) + Quant (N_bits=8)	**83**	***66.7***	*70.3 (+3.6)*	93.3
PCA (K=150) + Quant (N_bits=8)	**63**	66.1	*69.9 (+3.8)*	93.6
Autoencoder (K=200)	322	66.4	69.5 (+3.1)	92.9

* Compressed size of training + testing fMoW datasets (or original size, in the baseline case)
** Improvement over same input data with linear classifier

- ***Uniform quantization is lossless, even at 8 bits (vs 32-bit float)***
- *PCA models benefit most from deeper classifiers*
- PCA + quantizer performs/generalizes well, **despite large compression**
- EB has good compression and works well with a linear classifier
- Simple autoencoder model works but is universally outperformed

Future Work

- Experiment with depth, width, and output activation of autoencoder
- Train MLP autoencoder jointly as with the Entropy Bottleneck

References

[1] K. Ayush, B. Uzkent, C. Meng, K. Tanmay, M. Burke, D. Lobell, and S. Ermon, "Geography-aware self-supervised learning," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10181–10190, October 2021.
[2] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[3] Y. Yang and S. Newsam. "Bag-of-visual-words and spatial extensions for land-use classification," in Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, pp. 270–279, 2010.
[4] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in International Conference on Learning Representations, 2018.