

A Comparative Analysis of Machine Learning Methods for Image Compression

Jatin Vamshi Battu
Department of CSE
University College of Engineering
Osmania University

Oishi Nandy
Department of CSE
University College of Engineering
Osmania University

Abstract—Cameras are everywhere in today's world, from security cameras to smartphone cameras and webcams. All of these cameras produce an endless stream of images, which often need to be compressed for efficient storage and/or transmission. Image compression has been researched upon for many decades; however, in recent times, advances in machine learning have achieved great success in many computer vision tasks, and are now gradually being used in image compression. In this paper, we compare techniques involving Eigen Value based K-means clustering, GANs, CAEs, and the Gaussian Mixture Model using Bits Per Pixel(bpp) and PSNR(dB) as metrics

Keywords—Image compression, neural networks, machine learning, image quality.

I. INTRODUCTION

Digital media accounts for about 80% of the world's internet traffic, according to Cisco's Annual Internet Report [1]. With time, and the continuous increase in demand for digital media, this number is only expected to grow. Given the high volume of images being produced, stored and transmitted, image compression is of more vital importance now than it ever has been.

The objective of image compression is to decrease the redundant image data in order to store or transmit the data in a more efficient form. In essence, it aids in reducing the size (in bytes) of a digital image without significant degradation to image quality. In general, this is currently achievable due to the common characteristic of most images wherein neighboring pixels are correlated and therefore contain redundant information.

A. Traditional Approaches

Traditional methods of image compression like PNG, JPEG, and JPEG 2000, (which are currently the most widely used methods) use fixed transforms, i.e. Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (to first transform the image from its spatial domain representation to a frequency domain representation) together with the quantization and the entropy encoder to reduce spatial redundancies [2].

With recent advancements in the field of machine learning, which has conventionally been expected to drastically improve image compression, many new methods have been able to show potential in replacing traditional DCT/DWT methods with better, more efficient techniques. In this paper, we will be comparing some of these techniques.

In machine learning approaches, the structure is automatically discovered instead of being manually engineered. There are two fundamental steps in machine learning based image compression: selecting the most representative pixels as encoding, and colorization as decoding. The first step is essentially an active learning problem, whereas the second step is a semi-supervised learning problem [3].

B. Comparison Metrics

For the evaluation of compression approaches, we will be using BPP and PSNR. BPP (Bits Per Pixel) represents the average bits per pixel and is commonly used as a measure of efficiency in compression. We calculate the BPP based on this simple equation

$$BPP = J_s I_n$$

Here, J_s is the size in terms of bits, and I_n is the total number of pixels present.

PSNR (Peak Signal-To-Noise Ratio) is an important metric to measure the objective dissimilarity of the image. It is the ratio between maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. It is generally expressed in terms of the logarithmic decibel scale. We can calculate the PSNR using the following formula:

$$PSNR = 10 \log_{10} (MAX_I^2 / MSE) \text{ db}$$

Here, MAX_I is the maximum possible pixel value of the image, and MSE refers to the Mean Squared Error.

II. TECHNIQUES

A. Eigen Value based K-Means Clustering:

Although this method is not on the bleeding-edge of machine learning techniques for image compression, it does show some promise for replacing traditional methods of image compression. For years, the standard K-Means method has been used in image compression but in 2012, a method proposed by Somasundaram and Rani based on Eigen Values showed improvement over the traditional method [4].

K-Means is one of the most common partitioning techniques, and it has a large number of applications in the field of computing. The algorithm starts by choosing K initial seeds or centroids from the training data, either at random, or based on some heuristic measure. Next, it constructs a new partition by assigning each point to its closest initial centroid based on the distance measure, and the centroid of each set is

recalculated. The algorithm is repeated by the alternate application of these two steps until a convergence is reached.

In the aforementioned proposed method, the image is divided into 4x4 pixel blocks and the eigen value is calculated for every image block. The method then proceeds by partitioning the input image blocks into high and low eigen valued blocks. The classification is done based on a threshold which is the average eigen value of all the image blocks. Blocks with high eigen values contain all the principal components of the image whereas blocks with low eigen values consist of less detailed image components which can be subjected to high compression.

Next, clustering is performed on each partition with K1 and K2 blocks as initial seeds where K1 and K2 are the number of blocks having high eigen values in their respective partitions. The codebooks resulting from the high eigen and low eigen partitions are merged to form the global initial codebook. Finally, a last run of K-means with the global codebook as the initial codebook is done to construct the final global codebook.

B. Generative Adversarial Networks:

Mentzer et al. had proposed an extremely promising method [5] based on the usage of Generative Adversarial Networks (GANs), or more specifically Conditional Generative Adversarial Networks.

A GAN is a class of machine learning frameworks crafted by Ian Goodfellow and his colleagues in 2014. Here, two neural networks contest with each other in a ‘game’. Given a training set, this technique learns to generate new data with the same statistics as the training set. GANs have been hailed as one of the greatest achievements in the field of deep learning in recent years. The idea is to construct a generator and a discriminator [6]. The training purpose of the discriminator $D(\cdot)$ is to maximize its discriminative accuracy, while the training goal of the generator $G(\cdot)$ is to improve the authenticity of its reconstructed image as much as it can. In this training process, GAN adopts an alternating optimization method, and its objective function can be expressed by the following formula:

$$\min_G \min_D \mathbb{E}[\log D(x)] + \mathbb{E}[\log(1 - D(G(x)))]$$

In the Conditional GAN (CGAN), the generator learns to generate a fake sample with a specific condition or characteristics (such as a label associated with an image or more detailed tag) rather than a generic sample from unknown noise distribution. The method proposed by Mentzer et al. basically augments the neural image compression formulation with a CGAN.

When dealing with low bpp values, the existence of a GAN has a greater impact on the performance improvement because the insufficient bit allocation in the non-importance regions often occurs in the case of low bpp.

C. Compressive Autoencoders:

An Autoencoder is a type of neural network used to learn efficient data codings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal ‘noise’. Along with this, a reconstructing side is learnt, where the autoencoder tries to

generate a representation which is as close as possible to the original input. The idea of autoencoders has been popular for decades, and the idea of using them for image compression has been popular since the beginning.

Theis et al. had proposed a method [7] using compressive autoencoders (CAEs), which they defined as having three components; an encoder f , a decoder g , and a probabilistic model Q .

$$f: \mathbb{R}^N \rightarrow \mathbb{R}^M, \quad g: \mathbb{R}^M \rightarrow \mathbb{R}^N, \quad Q: \mathbb{Z}^M \rightarrow [0,1]$$

The discrete probability distribution defined by Q is used to assign a number of bits to representations based on their frequencies for entropy coding. All three components may have parameters and the goal is to optimize the tradeoff between using a small number of bits and having small distortion.

$$\underbrace{-\log_2 Q([f(\mathbf{x})])}_{\text{Number of bits}} + \beta \cdot \underbrace{d(\mathbf{x}, g([f(\mathbf{x})]))}_{\text{Distortion}}$$

Theis et al. used common convolutional neural networks for the encoder and decoder of the CAE. In their architecture, the encoder performs preprocessing, namely mirror padding and a fixed pixel wise normalization. Afterwards, the image is convolved and spatially downsampled, which is followed by three residual blocks. The decoder mirrors the architecture of the encoder. Instead of mirror-padding and valid convolutions, zero-padded convolutions are used. Upsampling is achieved through convolution followed by a reorganization of the coefficients. A convolution and reorganization of coefficients together form a sub-pixel convolution layer. Following three residual blocks, two sub-pixel convolution layers upsample the image to the resolution of the input.

D. Gaussian Mixture Model:

Gaussian mixture models are probabilistic models for representing normally distributed subpopulations within an overall population. Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Here, since the subpopulation assignment is not known, this constitutes a form of unsupervised learning. It is formulated by:

$$p_{\tilde{y}|\tilde{z}}(\tilde{y}|\tilde{z}) \sim \sum_{k=1}^K w^{(k)} \mathcal{N}(\mu^{(k)}, \sigma^{2(k)})$$

Cheng et al. proposed a method for learned image compression with discretized gaussian mixture likelihoods and attention modules [8]. Here, a neural attention mechanism equips a neural network with the ability to focus on a subset of its inputs. Typically, it is implemented as:

$$\mathbf{a} = f_{\phi}(\mathbf{x})$$

$$\mathbf{g} = \mathbf{a} \odot \mathbf{z}$$

In this method, discretized Gaussian mixture likelihoods are used to parameterize the distributions, which removes the remaining redundancy to achieve an accurate entropy model, and thus directly lead to fewer required encoding bits. In addition to this, a simplified version of an attention module into the network architecture has been adopted.

In the network architecture, residual blocks are used to increase a large receptive field and improve the rate-distortion performance. Decoder side uses subpixel convolution instead of transposed convolution as upsampling units to keep more

details. N denotes the number of channels and represents the model capacity. We use the Gaussian mixture model, thus requiring $3 \times N \times K$ channels for the output of the auxiliary autoencoder.

III. PERFORMANCE COMPARISON

To compare the performance of all the aforementioned methods, we used images from the Kodak Image Suite [9] which has long been used as a standard for compression testing, cropped to 256×256 . To compress the images, we have used the basic ideas of each of the aforementioned methods, and a machine equipped with an NVIDIA GeForce GTX 1050 Ti.



Figure 1: Sample images from Kodak Image Suite

The performance results of all four techniques is depicted in the following graphical representations:

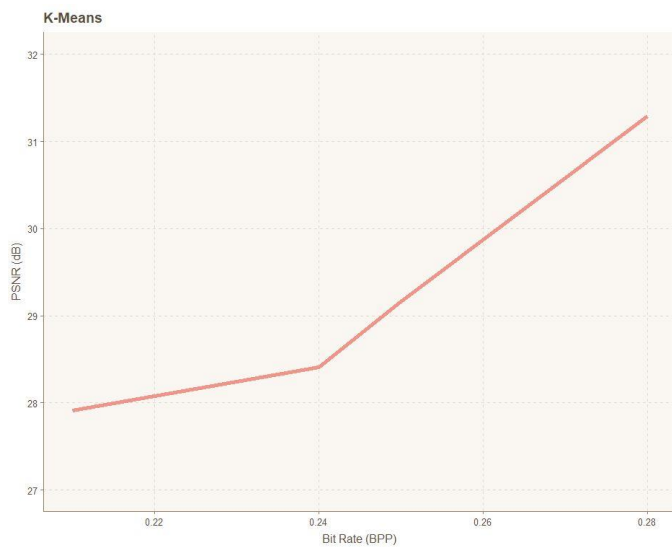


Figure 2: Performance Evaluation - Eigen Value based K-Means Clustering

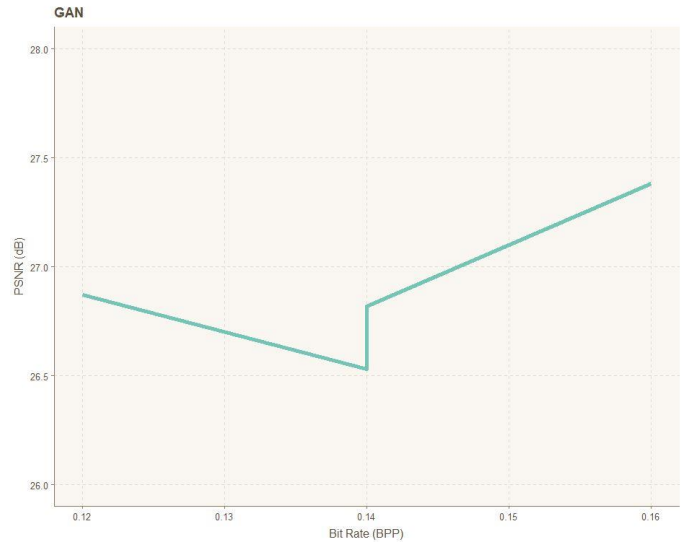


Figure 3: Performance Evaluation – Generative Adversarial Networks Autoencoder

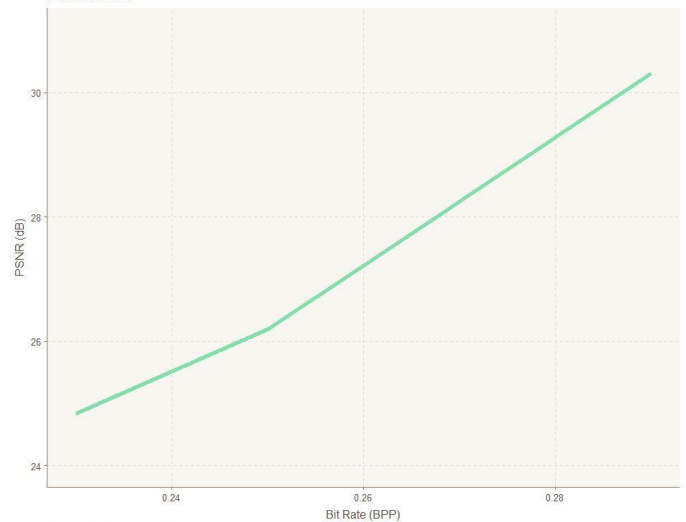


Figure 4: Performance Evaluation – Compressive Autoencoders

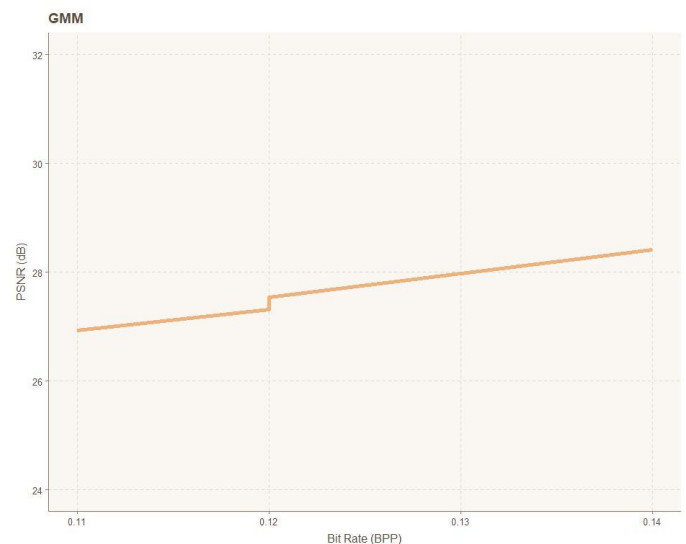


Figure 5: Performance Evaluation – Gaussian Mixture Method

IV. CONCLUSION

In this paper, we have compared machine learning methods for image compression including Eigen Value based K-Means clustering, General Adversarial Networks, Compressive Autoencoders, and the Gaussian Mixture Model. Machine learning methods inherently offer tremendous potential for increased efficiency in compression compared to the conventional methods that are popular today. While all of the aforementioned methods are novel and extremely adroit in their own right, we believe that GANs and CAEs have the most potential to improve image compression in the near future for image compression including Eigen Value based K-Means clustering, General Adversarial Networks, Compressive Autoencoders, and the Gaussian Mixture Model. Machine learning methods inherently offer tremendous potential for increased efficiency in compression compared to the conventional methods that are popular today. While all of the aforementioned methods are novel and extremely adroit in their own right, we believe that GANs and CAEs have the most potential to improve image compression in the near future.

REFERENCES

- [1] Cisco Annual Internet Report (2018–2023) White Paper, March 9, 2020.
- [2] Jayavrinnda Vrindavanam, Saravanan Chandran, Gautam K. Mahanti “A Survey of Image Compression Methods”, International Conference & Workshop on Emerging Trends in Technology, October 10, 2014.
- [3] Xiaofei He, Ming Ji, Hujun Bao, “A Unified Active and Semi-Supervised Learning Framework for Image Compression”, IEEE Conference on Computer Vision and Pattern Recognition, June 2009.
- [4] K. Somasundaram, M. Mary Shanthi Rani, “Eigen Value based K-means Clustering for Image Compression”, IJAIS, Volume 3– No.7, August 2012.
- [5] Fabian Mentzer, George Toderici, Michael Tschannen, Eirikur Agustsson, “High-Fidelity Generative Image Compression”, arXiv.org 10 Jul 2020.
- [6] Ian J. Goodfellow , Jean Pouget-Abadie , Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, “Generative Adversarial Nets”, Advances in Neural Information Processing Systems 27, NIPS 2014.
- [7] Lucas Theis, Wenzhe Shi, Andrew Cunningham & Ferenc Huszar, “LOSSY IMAGE COMPRESSION WITH COMPRESSIVE AUTOENCODERS”, ICLR, 2017
- [8] Zhengxue Cheng , Heming Sun, Masaru Takeuchi , Jiro Katto.” Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 7939-7948
- [9] Kodak Lossless True Color Image Suite, Download from <http://r0k.us/graphics/kodak/>