IMAGE COMPRESSION USING AI: BRIEF INSIGHTS INTO DEEP LEARNING TECHNIQUES AND AI FRAMEWORKS

Ravi Sankar Landu

SHARP Software Development India, Bangalore, India

Email: Lravisankar@yahoo.com

ABSTRACT

In this age of Internet and Cloud based applications, image compression has become all the more important, especially in the areas of Telemedicine, Satellite data etc. The images need to be compressed with high computation efficiency and regenerated with minimal losses. There are many techniques related to Image compression, mainly classified into Lossless and Lossy compression techniques. Many computer programs are developed for implementing these compression techniques. However, Deep Learning has been used for image compression since 1980s and has been adopted into most of the Artificial Intelligence (AI) platforms. This paper gives brief insights into some Deep Learning techniques, some AI platforms and different functions/methods available in those platforms.

Key Words: Image Compression, Deep Learning Techniques, Machine Learning, AI Frameworks.

1. Introduction

The process of representing the image with less number of bits by removing the redundancies from the image is called image compression [1] which is specified in terms of Compression Ratio (CR) or number of bits per pixel (bpp) termed bit rate. The main aim of image compression is to reduce the image file size so that it occupies less storage space, and less bandwidth for transferring over network. Necessity for Image compression comes from different applications and requirements such as:

- Applications are which are sensitive to power consumption, chip size and costs (mobile applications) [2]
- Applications with high intensive computing demands, such as GPS location, High Definition (HD) video recording and processing, Orthogonal Frequency-Division Multiplexing (OFDM) etc.[3]
- Applications having storage constraints (Dgital libraries, Databases, Medical imagery etc) [4]
- Applications requiring minimal network bandwidth (Satellite image transmissions).

2. Image Compression with Deep Learning

Deep Learning (DL) has been used for image compression since three decades and has led to development of different techniques

- Multi-layer Perceptrons
- Convolutional Neural Networks
- Generative Adversarial Networks

2.1 Multi-Layer Perceptrons

Multi-layer perceptrons (MLPs) include a hidden layer of neurons embedded between a layer of input neurons and a layer of output neurons. Multi-layer perceptrons with multiple hidden layers are useful for dimension reduction and data compression. Image compression with MLPs involves a unitary transformation of the entire spatial data.

The multilayer perceptron is used for transform coding of the image, in neural network applications. The network is trained for different number of hidden neurons with direct impact to compress ratio is experimented with different images that have been segmented in the blocks of various sizes for compression process [5]. The MLP algorithm for image compression incorporated conventional image compression mechanisms such as spatial domain transformation, binary coding and quantization into an integrated optimization task. In this algorithm, decomposition neural network is used to identify the optimal binary code combination for a given bit-stream output, but variable compression ratio is not achieved. The algorithm has been further developed with predictive techniques to estimate the value of each pixel based on the surrounding pixels. The MLP algorithm then uses backpropagation to minimize the mean square error between predicted and original pixels [6].

2.2 Convolutional Neural Networks

A convolutional neural network, also known as CNN or ConvNet, is a class of deep neural network that has been successfully applied to various computer vision applications, especially for analyzing visual images. CNN always contains two basic operations, namely convolution and pooling. The convolution operation using multiple filters is able to extract features (feature map) from the data set, through which their corresponding spatial information can be preserved. The pooling operation, also called subsampling, is used to reduce the dimensionality of feature maps from the convolution operation. Max pooling and average pooling are the most common pooling operations used in CNN [7].

Convolutional neural networks (CNNs) offer enhanced compression artifact reduction and super-resolution performance compared with traditional computer vision models. The convolution operations of CNNs allow them to determine how neighboring pixels correlate. Cascaded convolution operations mirror the properties of complex images. However, it is challenging to incorporate a CNN model throughout the image compression process, as it requires gradient descent algorithms and backpropagation, which are challenging to incorporate in end-to-end image compression.

2.3 Generative Adversarial Networks

Generative adversarial networks (GANs) provide a way to learn deep representations without extensively annotated training data. They achieve this by deriving backpropagation signals through a competitive process involving a pair of networks. The representations that can be learned by GANs may be used in a variety of applications, including image synthesis, semantic image editing, style transfer, image super-resolution, and classification [8].

A generative adversarial network (GAN) is a deep neural network consisting of two opposing generative network models. GAN image compression involves reconstructing a compressed image in a tiny feature space, based on the features from the input image. The main advantage of GANs over CNNs in terms of image compression is adversarial loss, which improves the quality of the output image. The opposing networks are trained together, against each other, enhancing the performance of the image generation model.

GAN is a new framework for estimating generative models prposed via an adversarial process, in which two models are trained simultaneously: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake [9].

3. Frameworks for AI-Based Image Compression

It is theoretically possible to code an entire image processing application yourself, but it is more realistic to leverage what others have developed, and simply adjust or extend existing software according to your own needs. Many existing frameworks and libraries provide models for image processing, many of them pre-trained on large data sets.

3.1 OpenCV

The Open Source Computer Vision (OpenCV) Library offers hundreds of machine learning and computer vision algorithms, with thousands of functions to support these algorithms. It is a popular choice, given its support for all the leading mobile and desktop operating systems, with Java, Python and C++ interfaces [10].

OpenCV contains numerous modules for image compression functions, including image processing, object detection and machine learning modules. You can use this library to obtain image data and extract, enhance and compress it [11].

3.2 TensorFlow

TensorFlow is Google's open-source framework that supports machine learning and deep learning. TensorFlow allows you to custom-build and train deep learning models. It offers several libraries, some of which are useful for computer vision applications and image processing projects. The TensorFlow Compression (TFC) library offers data compression tools [12].

You can use the TFC library to create machine learning models with built-in optimized data compression. You can also use it to identify storage-efficient data representations, for example of your images and features, which only minimally affect model performance. You can compress floating point tensors into much smaller sequences of bits.

3.3 MATLAB Image Processing Toolbox

Matrix Laboratory, or MATLAB, refers to both a programming language and a popular mathematical and scientific problem-solving platform. The platform offers an Image Processing Toolbox (IPT) containing various workflow applications and algorithms for processing, analyzing and visualizing images, and can be used to develop algorithms. MATLAB IPT enables the automation of image processing workflows, with applications ranging from noise reduction and image enhancement to image segmentation and 3D image processing. IPT functions often support generating C/C++ code and are useful for deploying an embedded vision system or for desktop prototyping [13].

Although MATLAB IPT is not open source, it does offer a free trial [14].

3.4 High-Fidelity Generative Image Compression

This high-fidelity generative image compression is a Github project, which leverages learned compression and GAN models to create a lossy compression system. This helps developers to experiment with the HiFiC code on Github[15]. This model is highly effective in reconstructing detailed textures in compressed images [16].

3.5 CompressAI

CompressAI is a platform that provides custom operations, layers, models, and tools to research, develop, and evaluate end-to-end image and video compression codecs. It uses pre-trained models and evaluation tools to compare learned methods with traditional codecs. Various models have been trained on learned end-to-end compression from scratch and re-implemented in PyTorch. Artificial Neural Network (ANN) based codecs have shown remarkable outcomes for compressing images. This framework currently implements models only for still-picture compression; however, it is believed to soon extend over to the video compression domain. CompressAI implements networks for still picture coding. It provides pre-trained weights and instruments to compare SOTA models with traditional image codecs [17].

4. Conclusion

This paper briefly discussed different techniques in image compression algorithms based on deep learning, including Multi-Layer Perceptrons, Convolutional Neural Networks, and Generative Adversarial Networks. Also, this paper provided a list of Artificial Intelligence based tools used to build image compression applications.

5. Future Scope

In depth information on the Image Processing libraries in each of the tools, OpenCV, TensorFlow, MATLAB, High-Fidelity Generative Image Compression and CompressAI, can be analysed. Also, different transformation methods, quantization techniques, Image evaluation (comparison of the output image with input image) models can be covered in the future papers.

References

1) *Gonzalez, R.C. and Woods, R.E., "Digital Image Processing". Pearson Education Asia, Second Edition, 2002.*
2) *H.Keding,M.Willems,M.Coors And H.Meyr, "FR1DGE:A Fixed-Point Design And Simulation Environment". Design Automation And In Test In Europe Proceedings Page:429-43 5,1998.*
3) *Chen Yang,Yizhuang Xie, He Chen And Yi Deng, "New Quantization Error Assessment Methodology For Fixed-Point Pipeline FFT Processor Design". IEEE International System-On-Chip Conference (SOCC), Pages: 299-305, 2014.*
4) *N. R. Wanigasekara; Shenghao Ding; Zhuangzhi Yan; Yongqin Zeng, "Quality Evaluation For PEG- 2000 Based Medical Image Compression". Engineering In Medicine And Biology,Annual Conference And The Annual Fall Meeting Of The Biomedical Engineering Society Embs/Bmes, Volume: 2,Pagcs: 101 9 - 1020, 2002.*
5) *Ivan Vilovic, "An Experience in Image Compression using Neural Network", University of Dubrovnik, 48th International symposium ELMAR, Multimedia Signal Processing and Communication, Pp. 95-98, 2006.*

6) URL:https://towardsdatascience.com/ai-based-image-compression-the-state-of-the-art-fb5aa6042bfa.
7) Shih-Chia Huang & Trung-Hieu Le, "Principles and Labs for Deep Learning", 1st Edition, 2021.
8) A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," in IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.
9) Goodfellow, Ian & et al., "Generative Adversarial Networks", Advances in Neural Information Processing System, 2014. doi: 10.1145/3422622.
10) URL: https://opencv.org/
11) Gary Bradski, Adrian Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, 1st Edition, September 2008.
12) URL: http://tensorflow.org.
13) Oge Marques, "Practical Image and Video Processing Using MATLAB", 1st edition, 2011.
14) URL: https://www.mathworks.com/
15) URL: https://github.com/Justin-Tan/high-fidelity-generative-compression
16) Fabian Mentzer et al., "High-Fidelity Generative Image Compression", arXiv:2006.09965, 2020.
17) J. Bégaint et al., "CompressAI: A pytorch library and evaluation platform for end-to-end compression research", arXiv:2011.03029, 2020.