Deep Neural Networks for Image Compression

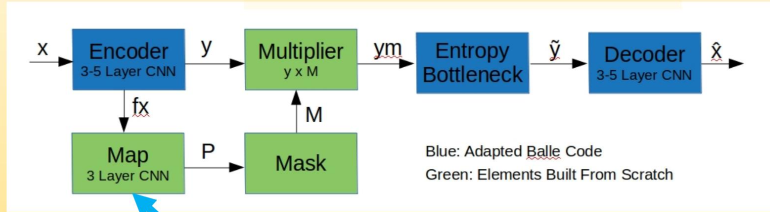
Brian Graham https://youtu.be/ph2ysJieqPA Jake Decoto
bgraham6@stanford.edu decotoj@stanford.edu

Abstract

Image compression standards, such as JPEG or HEVC, rely upon complex hand-tuned models to compress images. While these methods are generally effective, recent research has focused on machine-learning based techniques to supplement, or replace, these approaches.

We explore two existing state of the art deep learning algorithms. Capitalizing on advantages from each, we describe a new algorithm 'Balle-Li' with improved performance. This new algorithm was able to achieve similar distortion as that of our best hyperparameter selected version of Baseline Balle, while achieving superior compression performance.

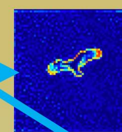
Balle-Li Autoencoder



Datasets

We utilized the Google Open Images dataset, which was designed for image classification and includes bounding box labels. When training our algorithms on AWS we used about 20k images of, on average, about a megabyte each. This dataset was randomly sampled during training, and we used patch sizes of 256 x 256 during training. To train our combined Balle-Li algorithm, we utilized this dataset without labels.

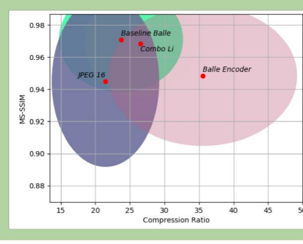
Importance Map Network

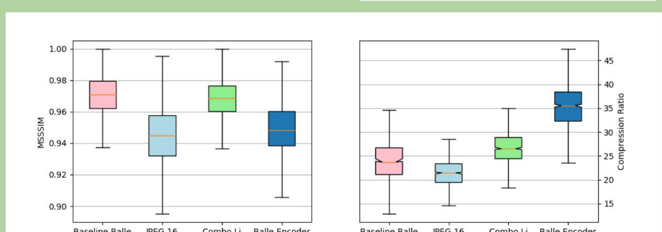


We heavily augmented Balle to build in an importance map network. The importance map enables a Convolutional Neural Network to learn how to allocate bitrate to local areas of an input image. For example, our network learns that the foreground bird is to be allocated greater bits than the background

Impressive Performance!

The Balle-Li algorithm achieves near industry-leading distortion performance (MS-SSIM) as compared to the Baseline Balle, while achieving 12% better compression ratio. 'Balle Encoder ' represents a re-worked version of Balle, in which a longer and more complex series of convolutional layers it utilized. This achieved much higher compression at the cost of increased distortion.





Loss – Rate Adjustment

$$J = \lambda * MSE + Rate \qquad Rate = 1/m * \sum_{i=0}^{m} log(likelihoods)/(log(numpixels))$$

Balle's algorithm utilizes a Gaussian Mixture Model to determine the rate term, in the classic 'Rate-Distortion Trade-off' in image compression. This Gaussian Mixture Model learns a likelihood that each latent variable – which is modeled as a Gaussian distribution – is correctly modeling the entropy of images. In contrast, Li's importance map network uses CNN layers to automatically learn a representation of entropy. This importance map is thus used as a rate term. We substantially modified the way that rate is determined in our Balle-Li algorithm by combining these approaches. For example, the input to Balle's 'entropy bottleneck' is now entirely distinct from that of Baseline Balle.

Future Work

We intend on leveraging Google's Open Images dataset to import known contextual information into our compression algorithm. For example, modern smart phones and social networks may automatically determine contextual information, such as a user in an image, or the user's friends, pets, and so on. Since this is done for free, we think adjusting the importance map to learn to allocate a greater bitrate for this contextual information – known to be important to users - makes sense. We made substantial progress on updating the code, but could not complete it in time. Based on our research, this appears to be the first such contemplated use of this dataset to train an image compression algorithm.