

Progressive Encoding-Decoding using Convolutional Autoencoder

Problem Statement

- Image Compression using Neural Networks
- Methods researched upon throughout the internship—
 - GAN – Generative Compression
 - RNN – LSTM Model
 - Residual Image, for compression

Method Used –

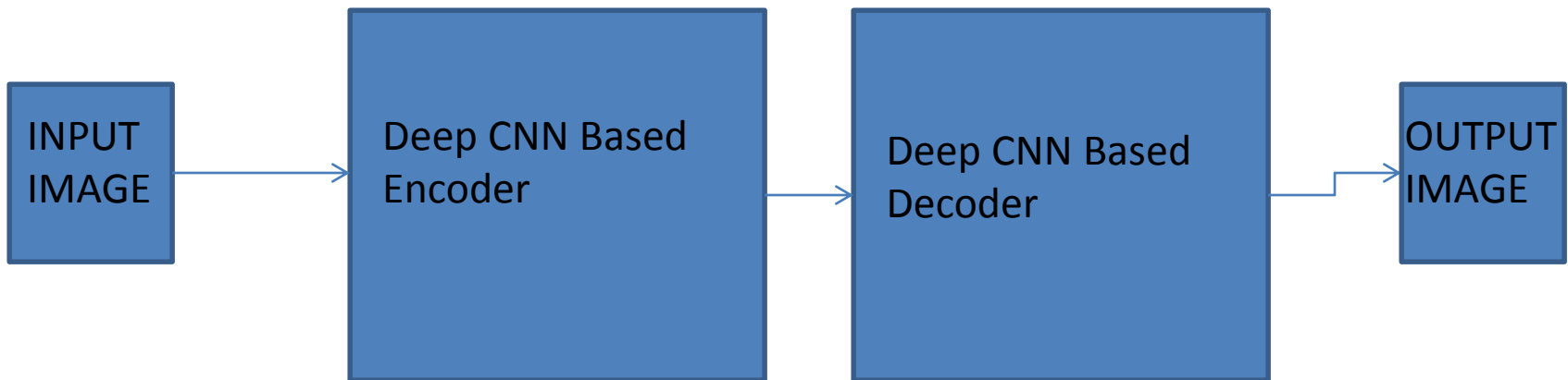
>Progressive Encoding-Decoding using CAE

References:

- <https://arxiv.org/abs/1703.01467>
- <https://arxiv.org/abs/1511.06085>
- <https://arxiv.org/pdf/1608.05148.pdf>
- <https://ai.googleblog.com/2016/09/image-compression-with-neural-networks.html>

Approach Used

- We have used a Deep Convolutional Auto-Encoder here, which progressively encodes and decodes the image.
- Progressive Encoding and Decoding basically means that once we specify a general architecture, we don't train the entire network **again** for different compression rates. We just train the newly added layers, w.r.t each compression rate.



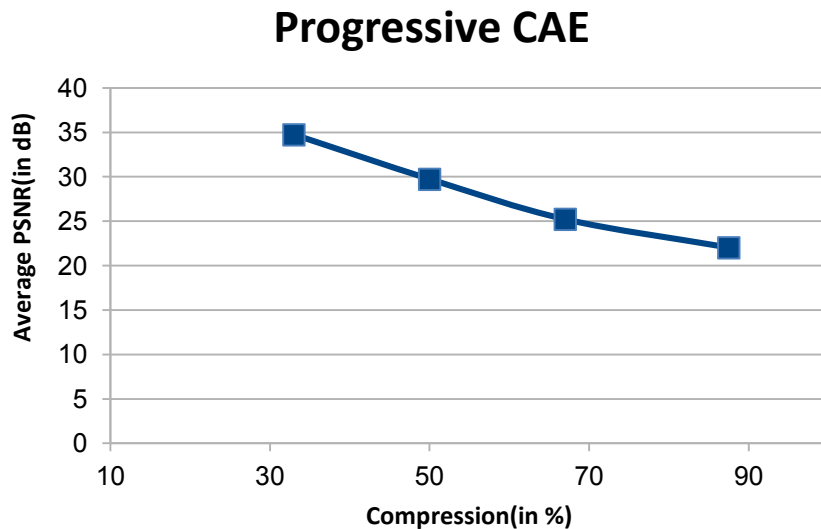
Implementation

- I used the CIFAR-10 and CIFAR-100 datasets combined together as both of them have 32×32 images and thus, my total dataset was 100000 images.
- I used 4 compression rates in this research: 33%, 50%, 67% and 87.5%
- The network for 33% compression has 5 convolution (including Pooling, and excluding Activation layers) and 5 deconvolution layers.
- The network for 50% has 9 convolution and 9 deconvolution layers.
- The network for 67% has 13 convolution and 13 deconvolution layers.
- The network for 87.5% has 16 convolution and 16 deconvolution layers.

P.S : The last network does not have 4 layers more than the previous trend, (which is observed as a general trend), because, on experimentation, it produced less quality images

Results

- Following are the graphs comparing JPEG (and JPEG-2000) with our research



Compression(in %)	Avg PSNR(in dB)
33	34.7377
50	29.73678369
67	25.21427181
87.5	22.03736997

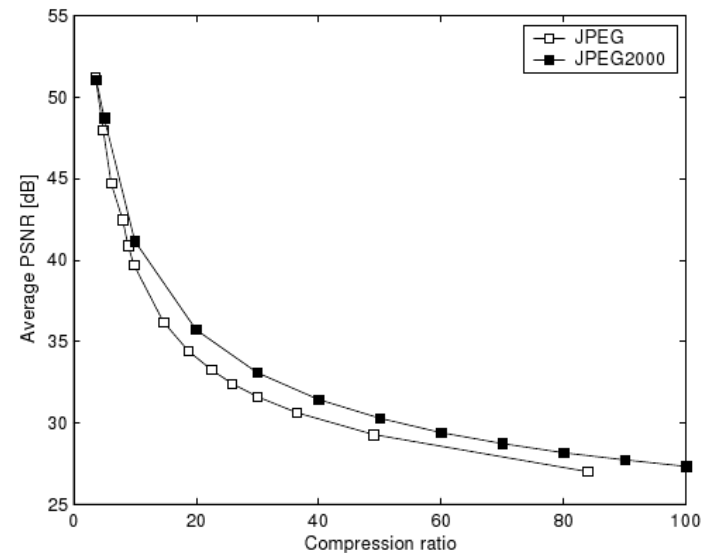


Figure 10. Average PSNR as a function of compression ratio for JPEG and JPEG2000.

Conclusions

- As we see, for relatively smaller compression ratios, the results are similar to JPEG but as we increase the compression rate, the quality of the algorithm we designed degrades.
- The number of filters and layers are variable and so, we can experiment and get similar and even better results than the JPEG compression.
- The algorithm was tested on a GPU 1080X and it took ~4 mins for the network with 32 layers to be computed.
- This is a great area of research, as image compression will be a major concern down the line.
- Other algorithms involving GAN's and RNN's look promising to work on and are found to exceed JPEG's compression results by a large margin.

- **Source: Samveed Desai, Research Intern, IISc Bangalore**