Team 8 – Final Project Report

Piyush Kumar Uttam
ee18btech11036@iith.ac.in

S. Sharath Chandra
es18btech11016@iith.ac.in

Rohit Reddy K
es18btech11010@iith.ac.in

G T Ramaneesh
es18btech11003@iith.ac.in

Abstract

Images are everywhere on the internet. For a random web surfer to view a single image, millions of bits have to be transferred over the congested internet traffic. Reducing the size of these images is a requirement given the modern standards. Thus, image compression techniques such as JPEG, WebP, AVIF, PNG etc were created. These techniques are so good and widespread now that an uncompressed format is now obsolete. But we can take it one step further by introducing Artificial Intelligence into image compression. The goal of this project is to perform lossy image compression using convolutional neural networks (CNNs), and achieve a better performance than image compression using codec.

1. Introduction

Image compression is the process of encoding an image in fewer bits than the original image for the purpose of storage or transmission. Uncompressed images require a huge amount of space for storage. For example, an uncompressed colour image of 1080p resolution (1920 x 1080) requires about 47MB. The main aim of compression is to minimize the storage space or bandwidth while preserving the resolution and quality of the original image.

Image compression is possible because images have redundancy. The redundancy can be in the form of spatial redundancy, which is the correlation between neighboring pixels. This occurs due to pattering or self-similarity within the image. Another form may be spectral redundancy which is caused by the correlation between coloured panels.

Image compression can be lossy or lossless. In lossless compression, the reconstructed image is identical to the original. However the compression achieved is not too much. In lossy compression, a high compression ratio can be achieved. However the quality of the image is reduced. Many are trying to come up with new lossy algorithms with the aim to improve the quality of the reconstructed image.We are also trying to achieve this by using neural networks.

2. Problem Statement

We aim to design a lossy image compression algorithm that is better than the existing codecs like JPEG, WebP, etc. We would like to achieve this by using deep learning neural networks. We will be designing 2 convolutional neural networks. The first CNN will aim to generate a compact representation of the image. This process is essentially a compression which is nearly(or completely) lossless. The output of this CNN will be processed using a standard codec like JPEG,WebP,etc. This process is a lossy compression. Then the second CNN will aim to reconstruct the original image from the output of the codec. This process will try to improve the quality of the image. Both the neural networks will be trained until satisfactory results are obtained. Thus The final reconstructed image will have a better quality for the same compression ratio, as compared to a standard codec.

3. Literature Review

The current image compression methods can be categorized into two segments, lossy and lossless image compression techniques. Deep learning adds another layer of novelty which is competitive to few of the existing best compression algorithms.

Below we discuss a few end to end image compression deep learning algorithms in detail which we have reviewed to work in our use case.

An End-to-End Compression Framework Based on Convolutional Neural Networks : Feng et. al.

Lossy Image compression : Lossy image compression refers to the algorithms which achieve high compression rates at the cost of loss of some details present in the original image, that is, it is impossible to perfectly construct the original image from its compressed counterpart. One such algorithm is the JPEG algorithm which suffers from huge compression artefacts at a high rate of compression.To tackle this problem, Feng et. at. proposed a dual network. The first network , which will take the image and generate a compact representation (ComCNN). The output of this network will then be processed by a standard codec (e.g. JPEG). After going through the codec, the image will be passed to a 2nd network, which reconstructs the original image through approximations made by the neural networks, hence getting rid of the compression artefacts. The authors called it Reconstructive CNN (RecCNN). Both networks are iteratively trained, similar to a GAN.

<u>Deep Image Compression via End-to-End Learning</u> : Liu et. al.

In this paper authors have proposed a deep residual network specifically, based image compression scheme that optimizes the end-to-end rate-distortion performance of image compression jointly. Overall structure consists of a forward encoder, a quantizer, a backward decoder, a rate-distortion optimization (i.e., rate estimation and distortion measurement) and a visual enhancement subsystems. Subsequently their method outperforms the existing BPG, WebP, JPEG2000 and JPEG compression algorithms as measured via multi-scale structural similarity (MS-SSIM), at the same bit rate.

4. Preliminary Results

[1] We have been working on our original problem statement. Out of the two neural networks, we were able to finish one. We used Keras to develop the neural network. For the compression part, as of now we have done image encoding using resizing and DCT. The output is passed through the reconstructor neural network. The neural network was trained on a dataset. Currently reconstructed images do not have better quality than JPEG images of the same size. We are working improving it. Some of the original and final reconstructed images are shown in the pictures below.

Original image:

Final Compressed image

The Compression ratio is 73.37%

Original image:

Final Compressed image:

The Compression ratio is 76.74%

[2] We are trying to recreate the results presented by Feng et. al.[1] from scratch as experimental code for the same is not available. The process involves two networks which are trained in sequential one by one manner as done in

training of GANs. This work is in progress however the model has been defined and only code for training is incomplete

We were planning to combine both [1] and [2] to get best possible results.

5. Proposed Approach

We tried using 2 neural networks to improve existing compression algorithms. After several tries, we failed to train a neural neutral that generates such compact representation of a given image. As the next best alternative, we used only one neural network to improve the quality of the compressed image by artifact removal.The dataset for this experiment was made using the RAISE Raw images dataset. We used the given uncompressed images and made a new dataset by randomly cropping the images into 720X720 patches. JPEG compression was performed on the prepared dataset. The quality factor of the compression was set to be 10%.

Our work is inspired by the previous work done on this problem. ARCNN is one such model which removes jpeg compression artefacts and improves the PSNR of compressed images. Overall, the AR-CNN consists of four convolution layers, namely the feature extraction, feature enhancement, mapping and reconstruction layer. A brief overview of ARCNN architecture is given in *image 1*.

The novelty in our approach is based on the addition of an ASPP module on the existing architecture. ASPP module was introduced in Deeplab semantic segmentation network and is used to increase the receptive field of the convolution operation without drastically increasing the number of learnable parameters.. The JPEG algorithm processes the image in 8X8 chunks.

As JPEG divides the given image into blocks and compresses each block individually, we lose a lot of data pertaining to a block
and its neighbours. By using atrous convolution, our model is able to learn patterns which correlate a block to its adjacent blocks and this results in improvement of quality.
We also added residual connection to prevent the problem of vanishing gradient.
The diagram for model architecture is as given in *image 2*.

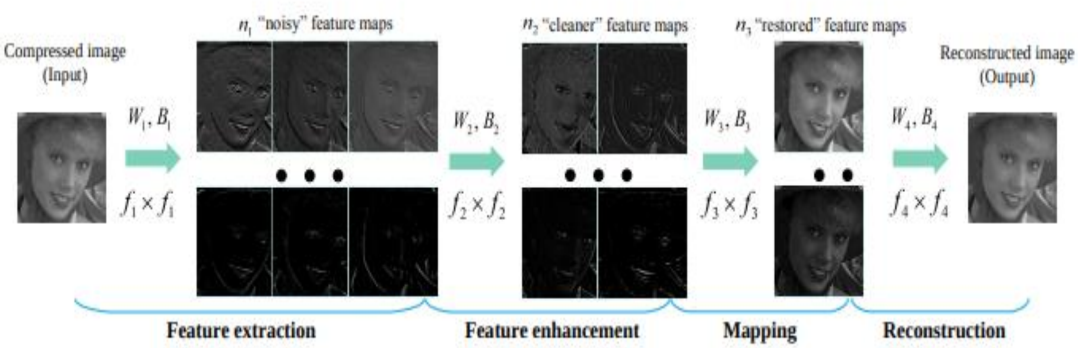


Image 1: ARCNN

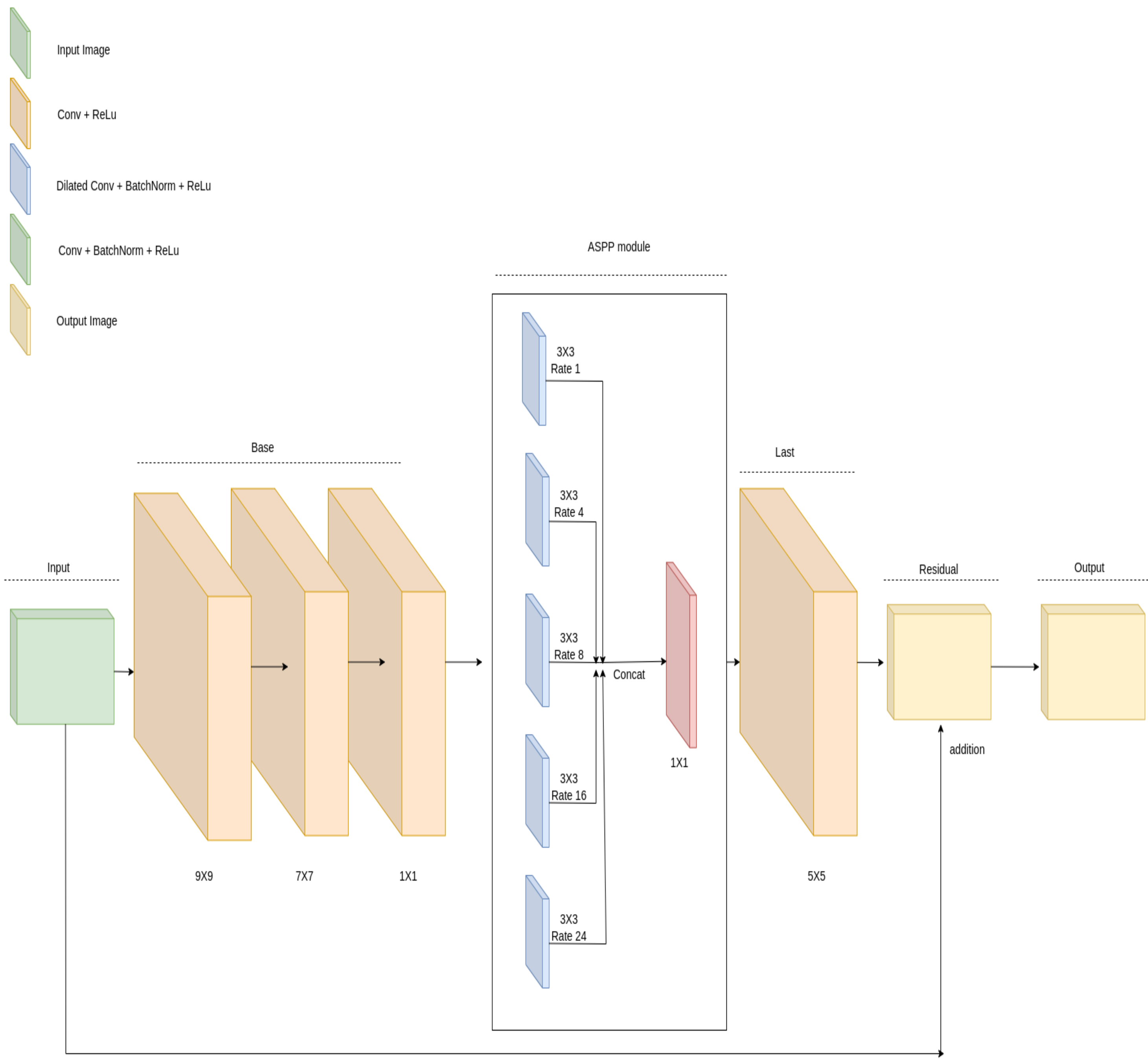
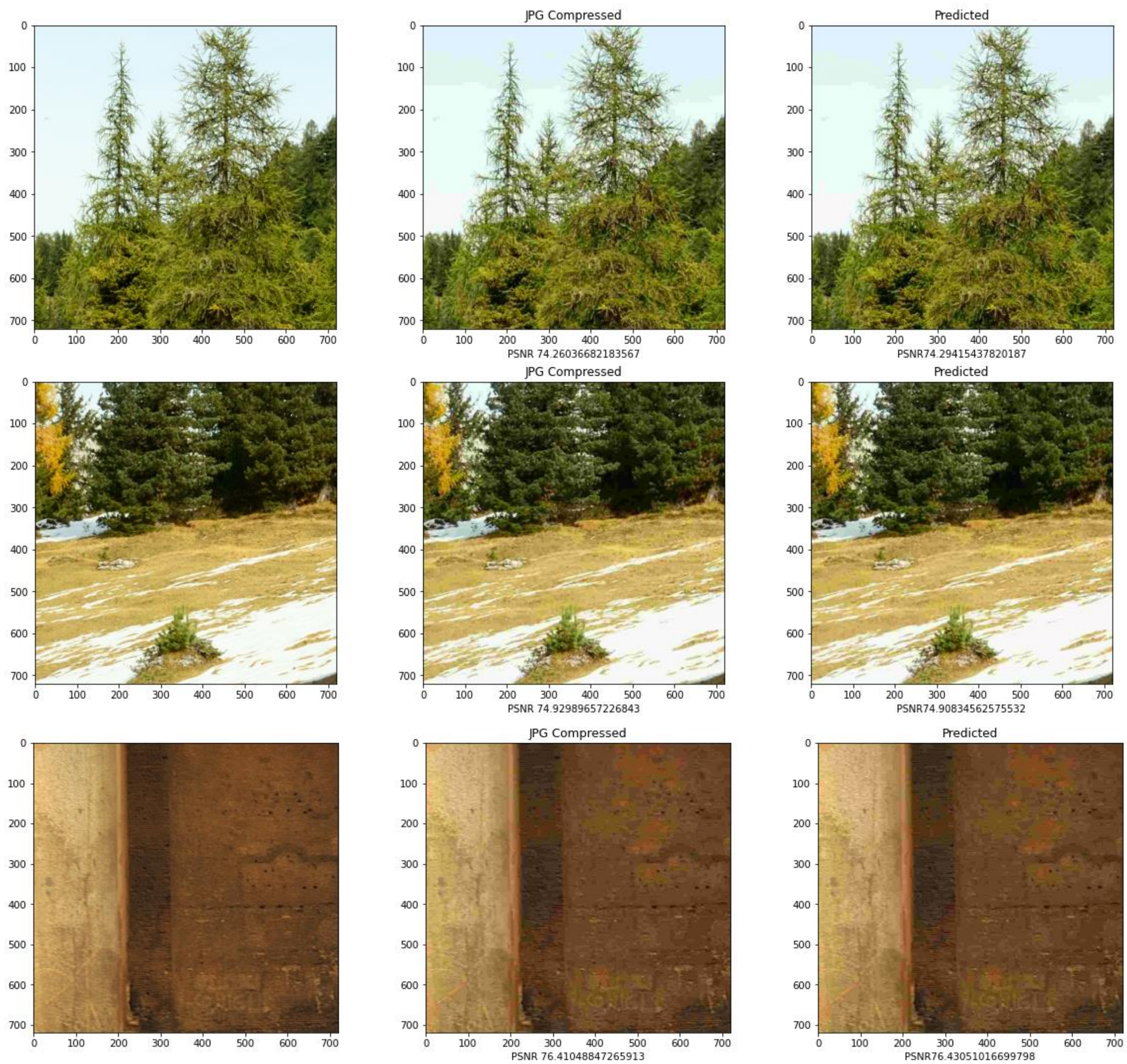


Image 2: Final CNN structure

6. Results and discussion

The proposed solution gave mixed results on the test dataset. Model was trained for 16 epochs and was stopped due to resource constraints and no convergence. Some of the resulting images are shown in *image 3*. It is evident from the calculation of PSNR that in some instances the model indeed improves the quality of the compressed image by reducing artifacts. In the current method, MSE was used as the loss function. Using MSE comes with its own disadvantages such as having little correlation between the structure of two images and the MSE between the two. Diagram in *image 4* illustrates this point perfectly. To counter this problem, in future works we can add SSIM loss instead of MSE loss. This new loss function might give better results than what we have now.



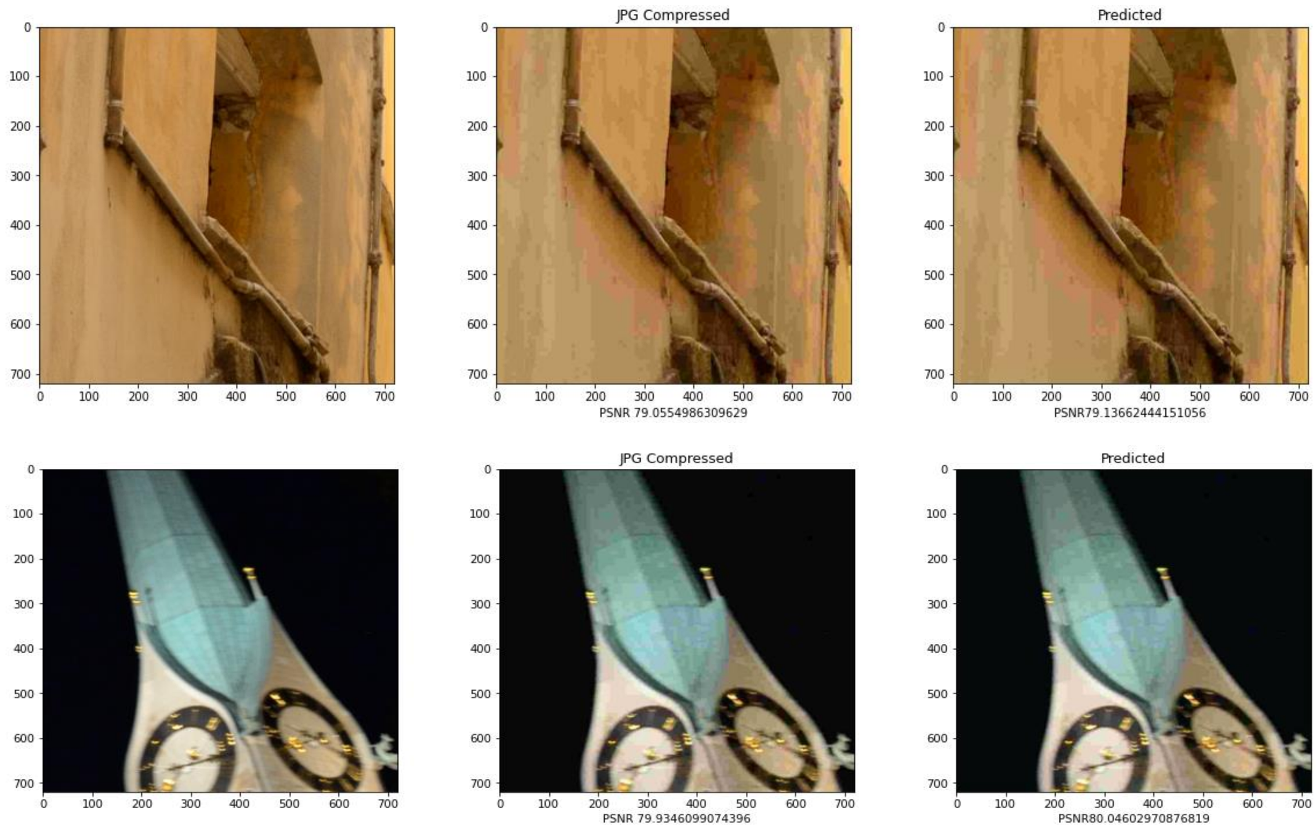


Image 3: Results

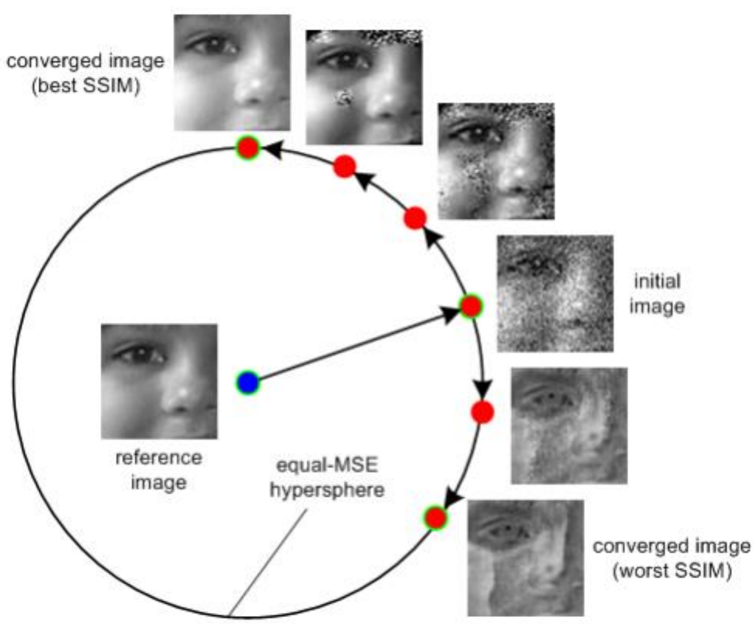


Image 4: MSE vs SSIM

7. Conclusion

In this project we carefully studied the JPEG compression algorithm and the artifacts induced by it during compression of image. We proposed a deep learning algorithm based on ARCNN but having a bigger receptive field using ASPP module and having residual connection. We further studied the use of loss functions and hypothesised the improvement of performance by using SSIM as loss function.

8. References

[1] Feng Jiang, Wen Tao, Shaohui Liu, Jie Ren, Xun Guo, Debin Zhao, Member, IEEE - **An End-to-End Compression Framework Based on Convolutional Neural Networks** - *arXiv:1708.00838v1 [cs.CV] 2 Aug 2017*

[2] Yuchen Liu , Zhixin Shu , Yijun Li , Zhe Lin , Federico Perazzi , S.Y. Kung - **Content-Aware GAN Compression** - *CVPR2021 12156-12166*

[3] Qing Jin, Jian Ren, Oliver J. Woodford, Jiazhuo Wang, Geng Yuan, Yanzhi Wang, Sergey Tulyakov - **Teachers Do More Than Teach: Compressing Image-to-Image Models** - *CVPR2021 13601-13611*

[4] Keras.io, Tensorflow Documentation

[5] ARCNN : https://arxiv.org/abs/1608.02778

[6] SRCNN : https://arxiv.org/abs/1501.00092

[7] Raise Raw Images Dataset : http://loki.disi.unitn.it/RAISE/