Challenges to convert Super Resolution GAN

Omar Faruk Riyad
1610634042
North South University

Arif Suhan
1610437042
North South University

May 2, 2019

Abstract

Deep learning approaches to single image super resolution have achieved impressive results in terms of traditional error measures and perceptual quality. We focused on the area where we could reduce the load over network and run on browser. To achieve this , we have faced a lot of challenges like loading the weight of pre-train model or running the network model in browser. As we are using pure vanilla javascript , we also have faced lack of libraries. When it's time to load weight and compute the network function , the browser cache crossed it's limitation.For solving these challenges , many solutions are crossed through our mind. After so many testing , we take the best solutions. Still some challenges are not solved.

1 Introduction

Reducing network load while transmitting data has been the great challenge of computer science since the dawn of the internet. Since the most transferred data over the network are images. If we could get high resolution images from low resolution could save the network load by significant amount. We wanted to try out a working mechanism that could achieve this particular goal while leaving the heavy duty on the user's computer. The state-of - the-art architecture ProSR is one of the best ways to retain image information to convert it to high resolution. The mechanism that we we tried was to include the computation on the browser side while the data sent by the host computer is lower than the data generated on the browser side. A dense model to work in the browser, because a lot of computation is needed, was actually a challenge. And they have used bicubic interpolation layer by layer to make the memory even higher For this particular model.

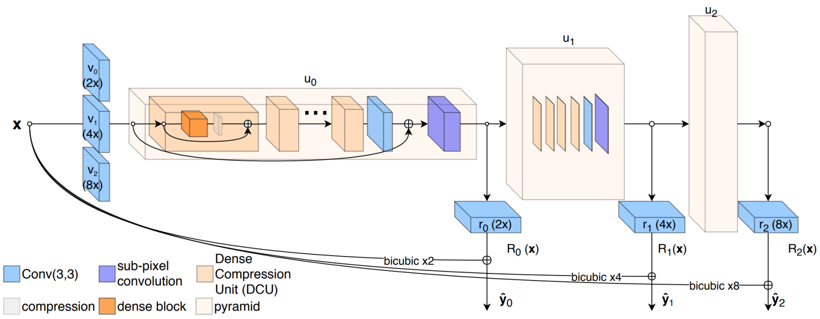
2 Model Description

2.1 Dense Net

Dense networks are the next step in increasing the profundities of deep convolutional networks. DenseNets are connected to networks. When CNNs go deeper, the problems arise. This is because the path from the input layer for information becomes so large that it can disappear before reaching the other part of the output layer (and the gradient is in the opposite direction). DenseNets simplify the pattern of connectivity among layers in other architectures. The architecture we have chose for the sake of exploration does have a dense block

2.2 ProSR GAN

Generative adversarial networks (GANs) have emerged as a powerful method to enhance the perceptual quality of the upsampled images in SISR.The model ProSR contained a dense layer and a dense compression unit. The model also used bicubic interpolation and addition with some of the layer output, for which it is one of the expensive model to perform computation with. [1]



3 Challenges

3.1 Converting Model

The first problem we got in order to get farther with the idea is to convert the model to JavaScript natively. We tried out a bunch of different options to convert the model but because of the model contained dense layers it was really difficult to put into code since the sequence of the network has to be followed. We found we could easily convert PyTorch model to Open Neural Network Exchange (ONNX) model, from the base of ONNX it performs operations ins Pure Basic

Language. We extracted the Pure Basic Model code from that model from which we found the underlying repeated network in the sequence of its weights. We found that there are 610 layers in the network and had about 7 operations which multiple different parameters. But in the end, we converted the model to native JavaScript.

3.2 Weight to JSON

The next challenge that we had to face is to convert weights and load it to the browser. Converting the model weights to JSON was complex but it was easily done. here again we used the ONNX model to get the weights from since our model network was generated from the ONNX. at first we worked with float64 data to have better precessions but the loading time was huge and it was expensive operation then we shifted to float16 sacrificing precision.

3.3 Engine Compatability

Firefox engine have Gecko,while Google chrome uses Blink.During the loading of Json file, Google Chrome shows time out. Sometimes it crashes so badly that browser has no responding.But in Firefox , it loads the Json file smoothly and fast.Never shows the any sign of crash.

3.4 Performing Operations Issue

Most of the used of the browser does not have GPU support since the whole idea of exploration was CPU based

3.5 Computation Issue

Computation with the model was higher than we expected. Since it had a huge amount of weight and layers to it and the model used bicubic and previous layer output farther down the network it had to store a lot of data to the memory. We tried some of the low res images to convert them to high resolution but we failed.

4 What to expect

If we could solve all these problems we could have faster data transmission for which the network could be reduced to more than 25% by assumption.

5 Evalution

5.1 Run Time

On runtime, we have seen that the model proSR usage 28GB of Memory to perform computation.

6 Conclusion

In our exploration, we have seen that computing in browser are high costly to run operation. With efficient highly rich library, hope this problems can be solved.Many other problems are also exists. The idea of using deep learing in the browser could open up new opprotunities in the realm of computer vision probelems.

References

[1] Brian McWilliams Yifan Wang, Federico Perazzi. *A Fully Progressive Approach to Single-Image Super-Resolution.* CVPR, 2018.