Machine Learning Engineer Nanodegree Capstone Project Proposal

Topic: Super-resolution imaging using Convolutional Neural Networks (SRCNNs)

Domain Background

Image processing techniques for enhancing the quality of images has been a topic of interest in the domain of Computer Vision for a long time. In general, two types of methods have been in practice, which are Spatial Domain Methods and Frequency Domain Methods [1]. Some of the methods listed in this paper include Histogram Equalization, Point Processing operations and other mathematical transformations on pixel data of the image to enhance the quality of gray scale images.

With the use of deep learning techniques, an image of lower picture quality can be enhanced to a higher quality image as is done by mapping low-quality phone photos into photos captured by a professional DSLR camera, by the students of ETH Zurich with the use of their self-created DPED dataset. [2]

One more field of research in image-enhancement domain is super-resolution imaging (SR) which involves techniques that enhance the resolution of an imaging system. Two conventional ways of doing this include Optical or diffractive methods and Geometrical or image-processing based methods. However, with the use of machine learning techniques like CNN, the relatively new method which has proved to be successful in this domain is the Super Resolution Convolutional Neural Networks (SRCNN), which I have chosen to work on as my capstone project.

Problem Statement

With the advancements in the field of Generative Adversarial Networks (GANs), these techniques are also implemented to achieve the goal of super-resolution [3]. However, I have chosen to proceed with the basic SRCNN model to achieve this task using conventional convolutional neural networks. A simple introduction to SRCNN can be seen in the medium article [4].

As my capstone project, I intend to train a deep convolutional neural network to achieve the goal of single image super-resolution based on the methodology as described in this paper [5].

The PSNR (Peak signal-to-noise ratio) is a widely-used metric for quantitatively evaluating image restoration quality and hence, can be used to evaluate our model.

Datasets and Inputs

As mentioned in the reference paper, I have chosen to use the famous T91 (91- image dataset) as an initial study to try and replicate the results mentioned in the paper which will help me gain an intuition about the working of the algorithm. Similar, to the paper, I use Set5 as the validation set for the SRCNN model and then to test the trained model, I would be using two other datasets like Set14 and Urban100.

The primary reason for choosing a smaller dataset for proof of concept is the computational constraints. However, later I plan to scale the process for larger datasets by using AWS services if possible. For this scaling up, I would be using one of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) datasets that contain thousands of images for training.

All the mentioned datasets for small-scale training can be found here [6]. The ILSVRC Datasets can be found on their website [7].

Solution Statement

While establishing the SRCNN model, the first step consists of using bicubic interpolation as a pre-processing step to increase the size of low-resolution image to the desired output size of the high-resolution image. Now, the image is scaled in size, but it is still of poor quality. So, in simple terms, the objective of our SRCNN model is to create a mapping function that can map the poor-quality image to the high-resolution image as given in the training set.

This process of “learning” the mapping function is done in three steps:

1] Patch extraction: Patches from the scaled low-resolution image are extracted which capture the key features of the image and form the set of feature maps used in a convolutional network training model.

2) Non-linear mapping: This step non-linearly maps each high-dimensional vector onto another high-dimensional vector. Each mapped vector is conceptually the representation of a high-resolution patch.

3) Reconstruction: The final step aggregates the high-resolution patch-wise representations to generate the final high-resolution image. This image is expected to be like the original high-resolution image X and the goal of the model is to minimize the error (difference between the obtained image and the original high-resolution image).

Benchmark Model

Normally, sparse coding methods were used for super-resolution process and hence, we shall consider the sparse-coding based super-resolution model as our benchmark model [8]. The up-scaling factor is 3 and the model achieves an average PSNR value of 31.42 dB

Evaluation Metrics

The primary metric of evaluation is the PSNR (Peak signal-to-noise ratio) which is the implementation of the mean squared error in the image processing domain. Higher the value of PSNR, lower is the mean-squared error of the model.

Another more suitable metric of evaluation is the structural similarity (SSIM) index which helps predict the perceived quality of pictures. Higher PSNR means more noise is removed but being a least squares result, it is slightly biased towards over smoothed results, that is along with the noise, some key features and textures of the image might be removed only to generate a high PSNR score.

SSIM, on the other hand, has quality reconstruction metric that considers the similarity of the edges (high frequency content) between the lower and higher resolution images. To have a good SSIM measure, an algorithm needs to remove the noise while also preserving the edges of the objects. SSIM is a "better quality measure", but it is more complicated to compute and hence, PSNR is the preferred evaluation metric.

Project Design

I have chosen to start by replicating the results mentioned in the paper [5] for the T-91 dataset (small-scale) training and then scale the results to larger datasets. The major challenge in this project is the design of the convolutional neural network model to get optimized performance. The exact details of the CNN model can be decided after rigorous experimentation on the number of layers, size of filters and other hyper parameter tuning processes.

After the implementation works for the results given in the referenced paper, I plan to modify and enhance the neural network architecture for better quality performance. Once I obtain an optimized model, I plan to try it out for different image sizes, filter sizes, and up scaling factors to emphasize the benefits of using SRCNN super-resolution models. The exact process cannot be finalized now as it will be an evolving process and can be finalized once I start working on the proposed project idea.

References

[1] https://arxiv.org/ftp/arxiv/papers/1003/1003.4053.pdf

A Comprehensive Review of Image Enhancement Techniques, Raman Maini and Himanshu Aggarwal

JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010

[2] http://people.ee.ethz.ch/~ihnatova/#dataset

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool.

"DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks",

in IEEE International Conference on Computer Vision (ICCV), 2017

[3] https://arxiv.org/pdf/1609.04802v5.pdf

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

CVPR 2017 • Christian Ledig • Lucas Theis • Ferenc Huszar • Jose Caballero • Andrew Cunningham • Alejandro Acosta • Andrew Aitken • Alykhan Tejani • Johannes Totz • Zehan Wang • Wenzhe Shi

[4] https://medium.com/coinmonks/review-srcnn-super-resolution-3cb3a4f67a7c

Review: SRCNN (Super Resolution)

[5] https://arxiv.org/pdf/1501.00092.pdf

Image Super-Resolution Using Deep Convolutional Networks

Chao Dong; Chen Change Loy; Kaiming He; Xiaoou Tang

[6] 91-image dataset and other validation and test datasets:

http://vllab.ucmerced.edu/wlai24/LapSRN/

[7] ILSVRC Datasets:

http://www.image-net.org/challenges/LSVRC/

[8] Benchmark Model where super-resolution is achieved using Sparse Coding techniques

Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE Transactions on Image Processing 19(11), 2861–2873 (2010)