Deep Image Prior

Aisha Meena
180070003

Divyansh Srivastava
180100040

Piyush Agarwal
180100080

***Abstract*—The Project has largely been a reading project which has been motivated by the desire of restoring degraded images. In this reference, we stumbled across the paper ”Deep Image Prior”[1] which promised to carry out image processing tasks without any training data set. This was done by handcrafting a neural network for the specific task and the specific image. In our attempt we have tried to understand the making of such architecture for two image processing tasks: denoising and super-resolution.**

I. Introduction

Image degradation is a major problem in the digital world where images are constantly captured, shared and stored. Degradation can happen at any stage of storing or transmitting. Clearly restoring such images is critical. Similarly, high-resolution is a critical task and finds its place in commercial websites to aid in shopping, medical imaging to aid surgeries and doctors.

As the models in the paper ”Deep Image Prior” stand, the restored images or the high resolution images do not surpass the state of the art techniques in denoising like CBM3D[2] or SRRESnet[3] for high-resolution but the method finds itself very handy when trained architectures cannot be used. A specific instance can be images of some novel virus whose predominant images are not available. The Deep Image Prior models however can be tweaked for the specific image to get useful higher resolution images. However, it is true that doing this in its own right is not easy and understanding the baseline convolutional neural network is tough.

A Few interesting things that come up in the paper were the consideration of how the hand-crafted network behaved as the prior for the image. Whatever information is stored about the task or the image is in the network and thus called a prior. The degraded image or the image to be super magnified features only in the error function which is used for gradient descent by the network provided the network has not been fined tuned for the image. Second thing was if gradient descent was run for long enough they converged to the respective input images so it was imperative to identify the right stopping point. Summarising, paper indicate that hand-crafted network architectures can be adequate to solve image restoration tasks up to certain levels of accuracy(without any explicit training).

II. Background and Previous Work

The work involves deep convolutional neural networks and a good understanding of the various concepts of CNNs is a must. Upsampling , Downsampling Layers have been extensively used. A good knowledge of how kernels work and various type of kernels(Lanczos, Box, Gauss) is very important.
A lot of work has been done on image processing and restoration tasks and there are state of the art methods but as the authors claim, *'this is the first study that directly investigates the prior captured by deep convolutional generative networks independently of learning the network parameters from images.'*
For Image Super Resolution some popular techniques are:

- LapSRN [4]- (Trained)
- SRRESnet- (Trained)

For image Denoising some popular techniques are:

- CMB3D
- NLM[5]

III. Data sets

As the model is not trained on any data set, there are no training data sets for the model. For extras we ran the code on some random images from the internet. Evaluation Study on explicit data sets could not be achieved due to lack of computational resources.

IV. Procedure and Experiments

Let x be the clean image, x_0 be the degraded image and x^* be the restored image. consider,

$$x = f_\theta(z)$$

From our prior knowledge of ML we know to get optimum x following equation need to be satisfied

$$x^* = \arg\max \mathrm{p}(\mathrm{x}_0 \mid x)$$

using the Bayesian rule ,

$$p(x \mid \mathrm{x}_0) = \frac{p(\mathrm{x}_0 \mid x)p(x)}{p(\mathrm{x}_0)} \alpha p(\mathrm{x}_0 \mid x) p(\underbrace{x}_{\text{likelihood}}) \quad \text{Prior}$$

$$\begin{aligned} x^* &= \arg\max \mathrm{p}(\dot{\mathrm{x}} \mid x) \\ &= \arg\min - \log p(\mathrm{x}_0 \mid x) - \log p(\chi) \\ &= \arg\min E(x; \mathrm{x}_0) + R(x) \ (2) \end{aligned}$$

$$x^* = \min_x E(x; x_0) + R(x)$$

$R(x)$ a regularizer. Now on taking R(x)=0 we get,

$$\theta^* = \underset{\theta}{\operatorname{argmin}} E(f_\theta(z); x_0), \quad x^* = f_{\theta^*}(z)$$

Our aim will be to achieve the above equation. The optimal theta, θ^* is obtained by using an optimizer such as gradient descent, starting from a randomly initialized x with random noise and theta.

A. *Super Resolution*

In SuperResolution, the input is a low resolution image $x_0 \in \mathbb{R}^{3\times H\times W}$ and a magnifying factor t such that the output is a high resolution image $x \in \mathbb{R}^{3\times tH\times tW}$. The governing loss function for the task is:

$$E(x, x_0) = ||d(x) - x_0||^2$$

where $d(.) : \mathbb{R}^{3\times tH\times tW} \rightarrow \mathbb{R}^{3\times H\times W}$ is a downsampling operator and in this case a Convolutional Neural Layer with a Lanczos Kernel[6] was used.
We'll talk about how super-resolution has been implemented in the paper.As pre-processing steps, the original image is trimmed at the borders so that both the height and width of the image is divsible by 32. (This is because of 5 downsampling layers in the magnifying neural net which reduces height, width by a factor of 2) Note that a Low Resolution image is not taken as input but the image with the original specifications is reduced by the factor of desired magnification and this resized image is then used to generate the desired high resolution image.

1) Model Results: The above-discussed methods solve the problem of image degradation to a great extend. We were successfully able to obtain a clear version of the degraded image.

Fig. 1: Degraded input image

2) Evaluation: The evaluation metric was PSNR value for the framed dataset and it was compared against trained CNN models like SRRESnet and LapSRN.

B. *Denoising*

Denoising the process of obtaining a clear image, x from a noisy image, x_0.
consider a model, $x_0 = x + \epsilon$ where, distribution ϵ is the noise and may optionally be incorporated into the model, though this process works well in blind denoising. The loss function:

$$E\left(x, x_0\right) = \left\|x - x_0\right\|^2$$

Fig. 2: Ground Truth

Fig. 3: Clear output image

is used and to obtain θ^* (i.e. optimal theta) we optimize:

$$\min_\theta \left\|f_\theta(z) - x_0\right\|^2.$$

The theta that we are optimizing over represents the weight and the structure of the model. We assume that some prior of the image is present inside the architecture of the model and and the parameters of the model.
We take some random noise in the form of an image (z) and feed that into the model. The model returns us some output that is the map of the input (z). The output is $f_\theta(z)$ for some set of params that we call θ. We optimize over this set of parameters (θ) such that the map $f_\theta(z)$ more closely resembles the noisy image, i.e., the entropy loss between the output of the model and the input noisy image is minimized.

1) Model Results: Below we can see an image in three forms: i) ground truth (fig 4), ii) noisy image (fig 5) and iii) output generated by the model (fig 6).

2) Evaluation (Losses and PSNR): As the number iterations go on, the entropy loss goes on decreasing and the PSNR (compared with the ground truth) of the model output rises, attains a maximum and then falls (fig 7). When the PSNR is at maximum, we can say that the amount of signal compared to the amount of noise in the output is very large. So, we can stop our iterations when the psnr value attains the maximum.

Here, we can see that the PSNR peaks around 2500 iterations for this particular image data. So, we preferred to stop our iteration at 2500 iterations.

Fig. 4: Ground Truth

Fig. 5: Noisy Image

Fig. 6: Model output

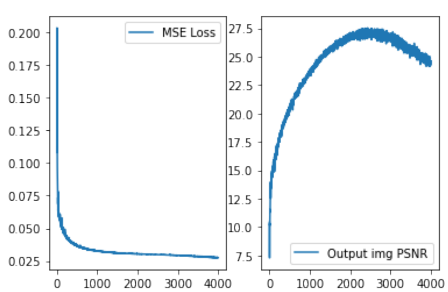

Fig. 7: Variations of MSE Loss and PSNR score with numbe of iterations

V. Network Architecture

An HourGlass Encoder-Decoder Architecture is usec In this architecture there are a number of downsamplin blocks which are then followed by upsampling blocks The upsampling technique used was bi-linear, and for downsampling convolutional neural layers with stride length of two were employed. Batch Normalization layers were used up to speed up gradient descent for optimization. For padding reflection padding was used. In both the tasks the network architecture, activation functions were identical except that the number of iterations were different.

The Activation function used is LeakyRelu[7] which is a modified form of Relu. (Function and graph is provided below) (On the original Paper experiments were done with a number of activation functions like Relu, LeakyRelu and even Swish).

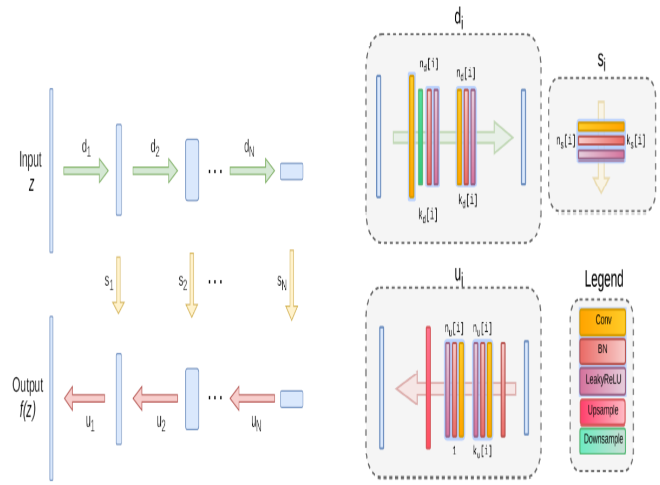

Fig. 8: Architecture

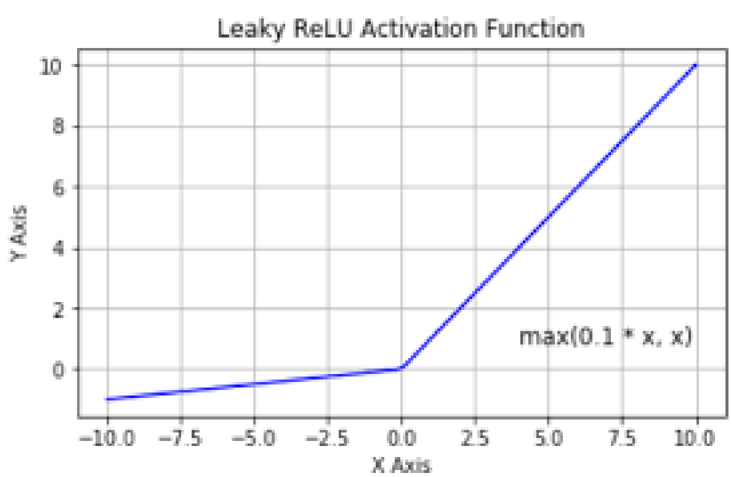

Fig. 9: Activation Function

VI. Conclusion

Commenting on the Deep Image prior paper, it shows how some aspects of images are contained in the network architecture which makes it possible to generate a clear image from a degraded image. However, the method itself is not rooted in wide applications as also commented by the author due to the nature of the method. (Need to fine tune the architecture of the neural network for each image). This is in line with the author noting that the main aim of the Deep Image Prior is to show the connection between image and networks and not to use them for image processing tasks.
As for ourselves, it definitely was a difficult read. With limited knowledge in Deep Neural Networks, having to tackle various upsampling, downsampling ideas was tedious and we had to start from zero. With limited experience in Pytorch, even understanding the code took quite a lot of time for us let alone make changes and do some of our own things which we could have hoped for. All in all for us it was a good learning experience.

A. Future Work

There are clearly a lot of things that we still need to do. Understanding the code is done, but still we have to reach there where we can modify the architecture and fit it for a specific image and do our own experiments. The author has carried out the task of super-resolution with only factors of 4 and 8. So we can attempt for higher factors like 16 and 32 and even smaller ones like 2 and 3.

VII. Statement of contributions

Aisha Meena: Studied the paper, ”Deep Image Prior” by Dmitry Ulyanov, Andrea Vedaldi,and Victor Lempitsky. Studied super-resolution from the papers given as a reference in Deep Image Prior. Wrote the report. Also, contributed to one-third of the video explaining the overall procedure and utility of the project.
Piyush Agarwal: Studied the paper, ”Deep Image Prior” by Dmitry Ulyanov, Andrea Vedaldi,and Victor Lempitsky. Analyzed the pre-existing code and rewrote it for super-resolution. Performed super-resolutions on images. Made one-third of the video explaining the super-resolution part of the experiment.
Divyansh Srivastava: Studied the paper, ”Deep Image Prior” by Dmitry Ulyanov, Andrea Vedaldi,and Victor Lempitsky. Analyzed the pre-existing code and rewrote it for Denoising. Performed denoising on images. Also, contributed to one-third of the video explaining the denoising part of the experiment.

Acknowledgments

We would like extend our gratitude to Prof. Amit Sethi and the TA team of the spring 2020 course EE - 769, Department of Electrical Engg. for inspiring us with this idea and guiding us throughout the duration of the project on different aspects.

References

[1] Dmitry Ulyanov, Andrea Vedaldi,and Victor Lempitsky, Deep Image Prior, 2017.
[2] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing 16(8), 2080-2095 (2007) 6, 7, 20
[3] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J.,Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In:CVPR. IEEE Computer Society (2017) 1, 2, 8, 9
[4] Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. IEEE Computer Society (2017) 2,8.
[5] Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proc. CVPR, vol. 2, pp. 60-65.IEEE Computer Society (2005) 6, 7, 20
[6] Turkowski, K.: Filters for common resampling-tasks.Graphics gems pp. 147-165 (1990) 19
[7] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: CVPR, pp. 1026-1034. IEEE Computer Society (2015) 14
[8] Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proc. ICCV, pp. 349-356 (2009) 9, 20