Deep Image Prior

Paper Link: https://arxiv.org/pdf/1711.10925.pdf

The Three Musketeers (Team 10)

1. Aman Khandelwal (2022201010)
2. Nikhil Khemchandani (2022201042)
3. Piyush Singh (2022201032)

Problem Statement

- ConvNets are commonly trained on massive image datasets.It's possible to believe that their exceptional performance results from their ability to learn realistic data priors from examples.

- However, this explanation is not sufficient because not all image priors need to be acquired through data learning.A significant portion of image statistics can be captured by the structure of generator ConvNets, without any learning.This is particularly relevant for addressing specific image restoration tasks.

- In these tasks, the image prior must compensate for the loss of information incurred during degradation.For such tasks, the statistics required can be obtained from the ConvNet structure itself, rather than from data learning.

Solution Proposed in the Paper

The proposed solution in the "Deep Image Prior" paper presents a new paradigm for image restoration tasks that exploits the inherent structure of deep neural networks, and provides an alternative to traditional data-driven approaches such as..

1. **Denoising**: the process of removing noise from an image.
2. **Inpainting**: the process of filling in missing or damaged portions of an image, for example, when part of an image has been obscured or removed.
3. **Super-resolution**: the process of increasing the resolution or size of an image beyond its original dimensions, often used to improve the clarity and quality of images.

Scope of the Project

We have implemented two applications from the paper :

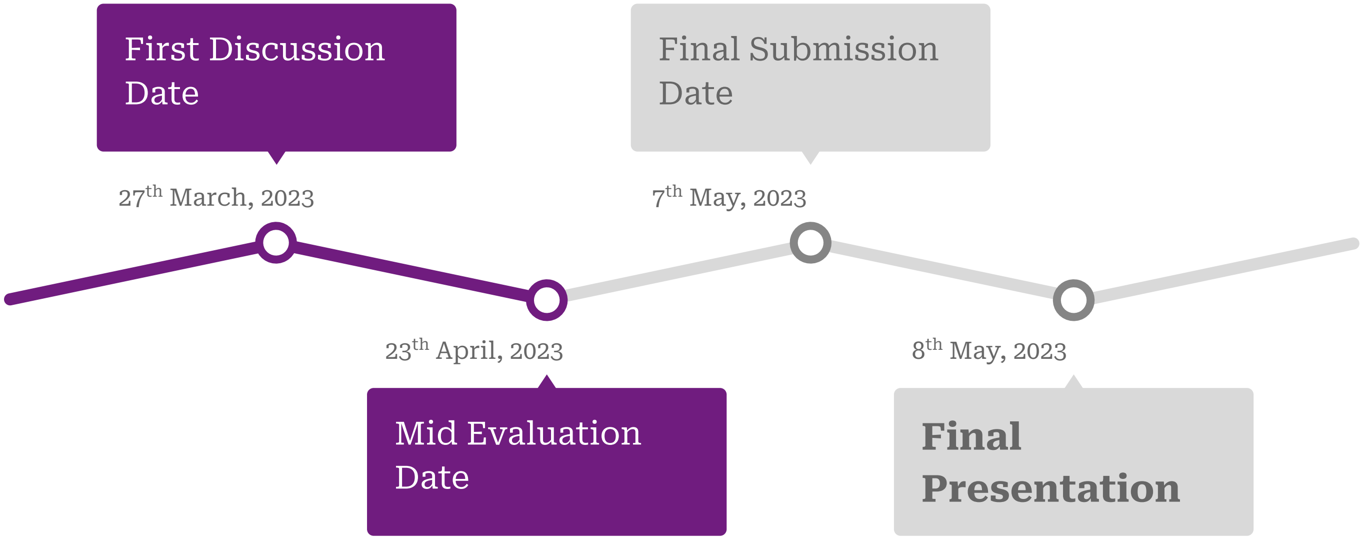
1. Denoising
2. Inpainting

    2.1. Text Inpainting

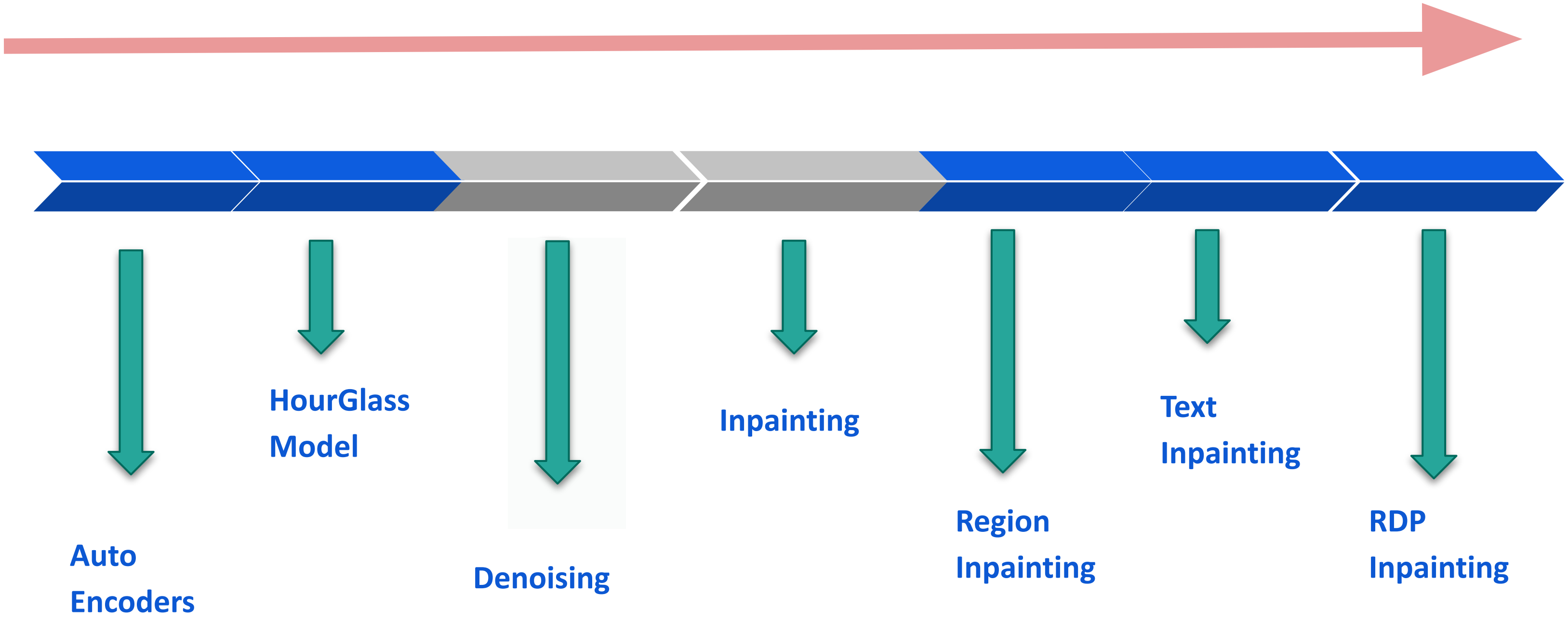
    2.2. Randomly Damaged Pixels Inpainting(RDP)

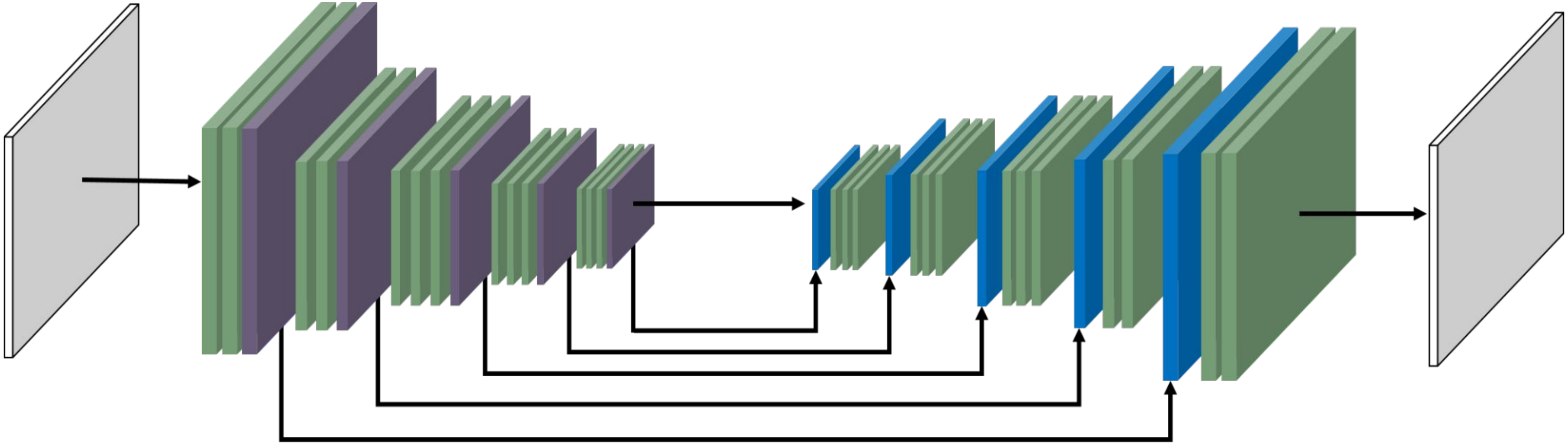
    2.3. Region Inpainting

Timelines


First Discussion Date
27th March, 2023
Mid Evaluation Date
23th April, 2023
Final Submission Date
7th May, 2023
Final Presentation
8th May, 2023

Implementation Details





<u>HOURGLASS MODEL</u> -

- ➢ The network consists of a contracting path and an expansive path, which gives it the u-shaped architecture.
- ➢ The contracting path is a conventional convolutional network, consisting of convolutions applied repeatedly, each followed by a rectified linear unit (ReLU) .
- ➢ The expansive pathway combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path.

Deep Image Prior Step By Step

- $\hat{x}$ – Corrupted image (observed)

1. **Initialize** z
   - For example fill it with uniform noise U(-1, 1)
2. **Solve**

$$\arg\min_{\theta} E(f_{\theta}(z); \hat{x}))$$

   - With your favorite gradient-based method

$$\theta^{k+1} = \theta^{k} - \alpha \frac{\partial E(f_{\theta}(z); \hat{x})}{\partial \theta}$$

3. **Get the solution**

$$x^{*} = f_{\theta^{*}}(z)$$

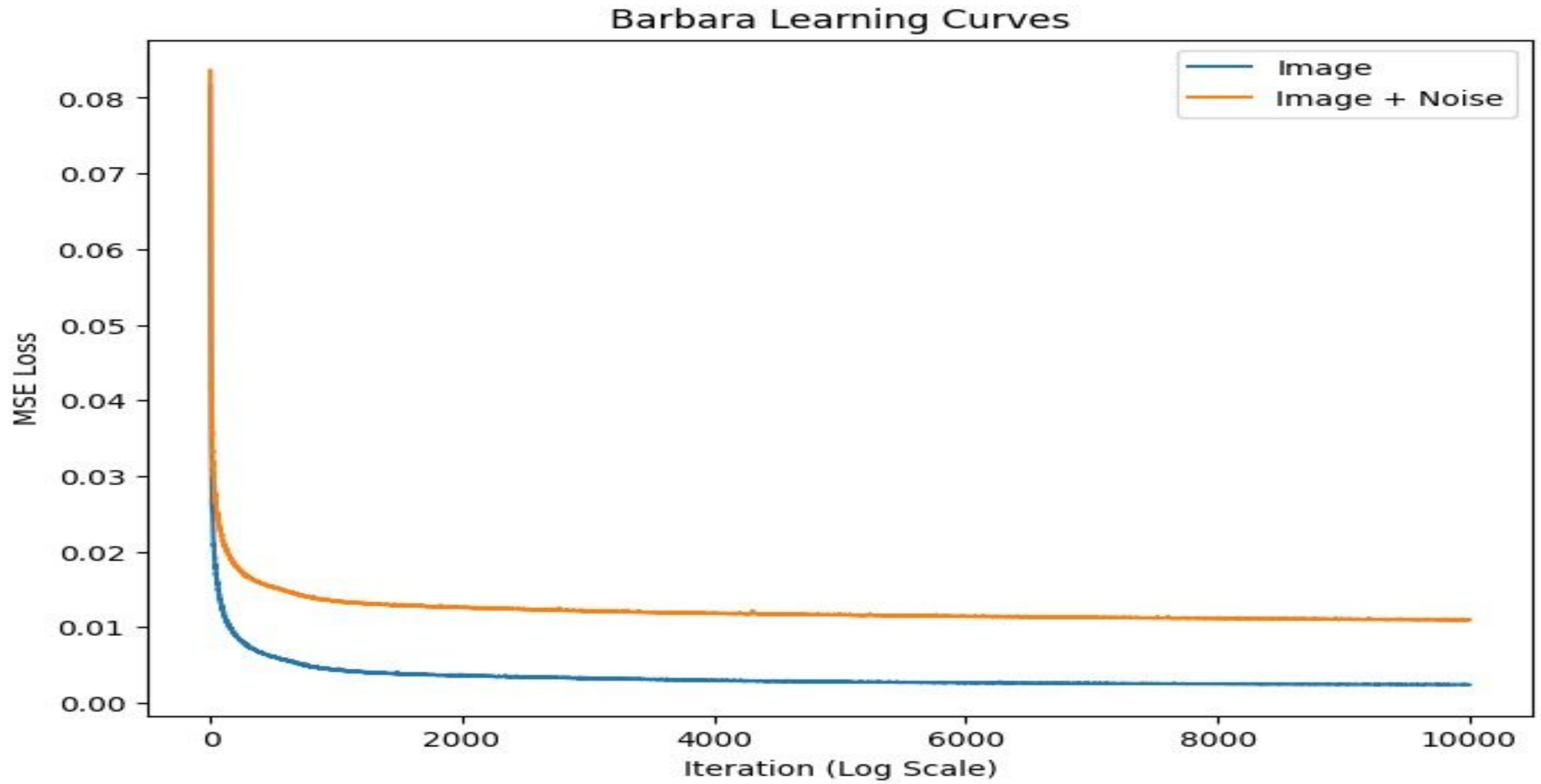
Objective Function

- x – Clean image
- $\hat{x}$ – Corrupted image (observed)
- m – Binary mask

Objective: $\arg\min_{\theta} E(f_{\theta}(z); \hat{x}))$

- **Denoising:** $E(x; \hat{x}) = \|x - \hat{x}\|^2$ **Needs early stopping!**
- **Inpainting:** $E(x; \hat{x}) = \|(x - \hat{x}) \cdot m\|^2$

Learning Curves for the Reconstruction Task (Experiment)



Experiments

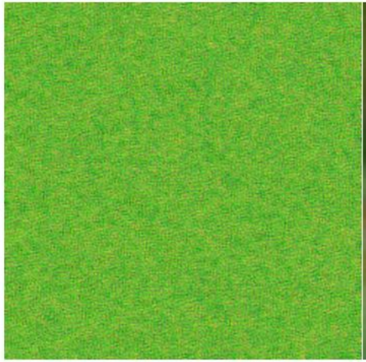
Original Image

Noisy Image

Reconstructed Image (8000 Iterations)

Mean Squared Error: 0.008465769766482794 (Between Original Image and Reconstructed Image)

Denoising Results



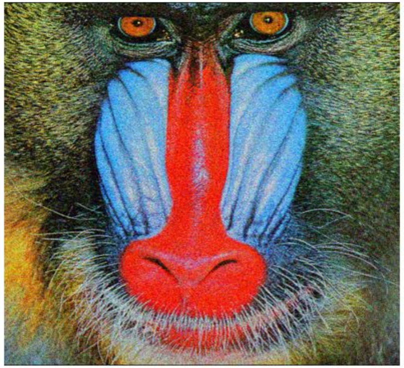
Random Noise

500 iterations

1500 iterations

2500 iterations

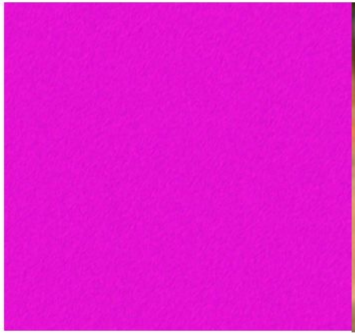
10000 iterations



Noisy Image

Original Image

Denoising Results

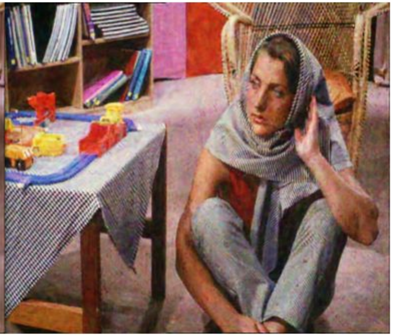


Random Noise

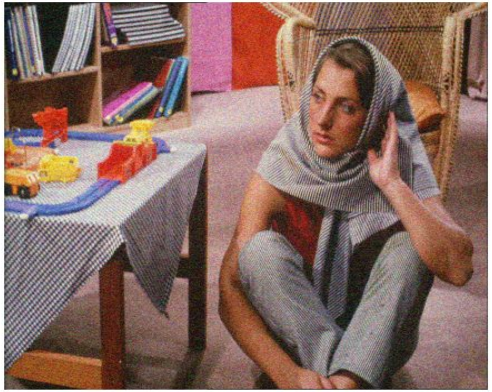
500 iterations

1500 iterations

2500 iterations



10000 iterations



Noisy Image

Original Image

Denoising Results

Random Noise

500 iterations

1500 iterations

2000 iterations

10000 iterations

Noisy Image

Original Image

Text Inpainting Results

Noisy Image

500 iterations

1500 iterations

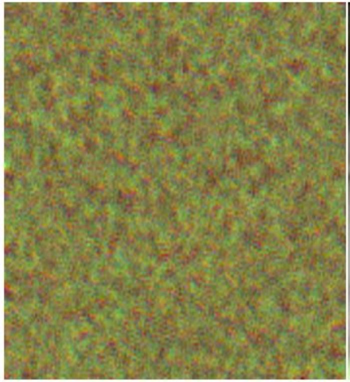
2500 iterations

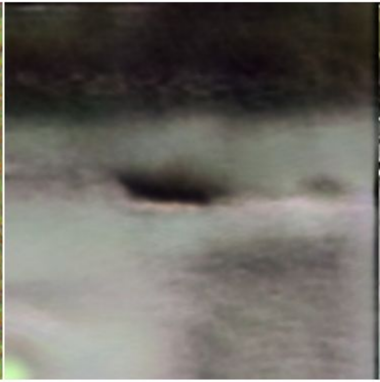
8000 iterations

Corrupted Image

Original Image

Text Inpainting Results

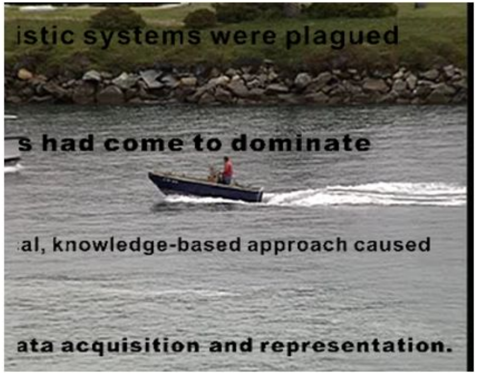

Random Image


500 iterations


1500 iterations


2500 iterations

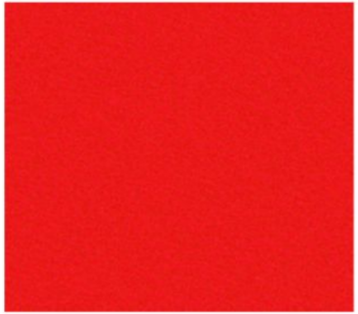

8000 iterations



Corrupted Image

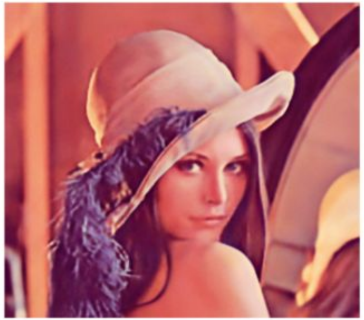

Original Image

Randomly Damaged Pixels Inpainting Results

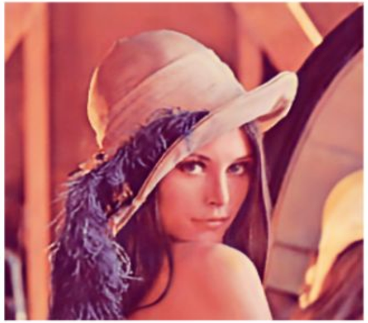


Random Noise

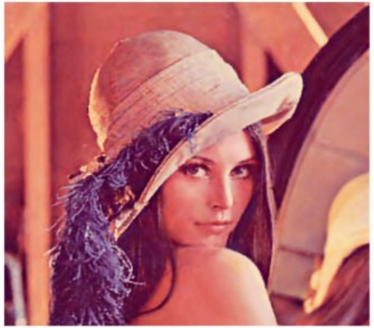
500 iterations



1500 iterations

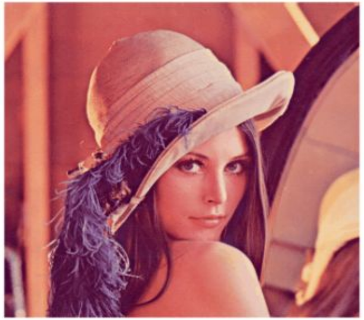


2500 iterations



8000 iterations

Corrupted Image

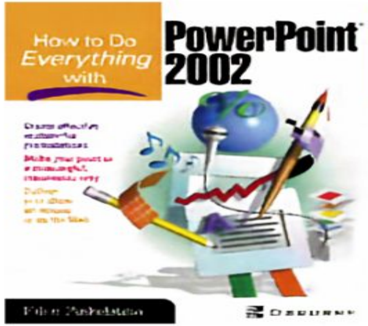


Original Image

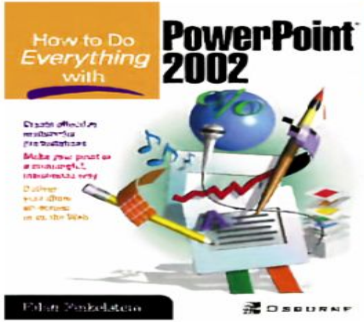
Randomly Damaged Pixels Inpainting Results

Random Noise

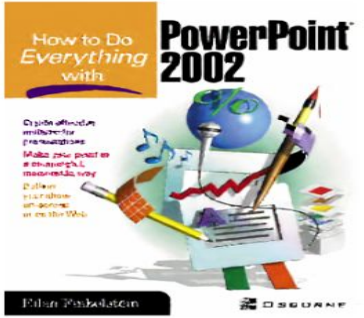
500 iterations



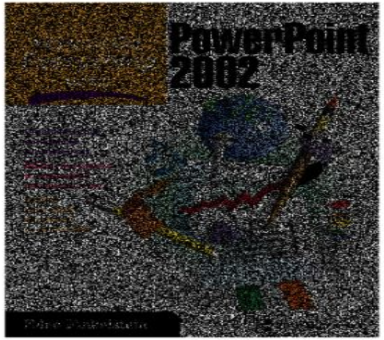
1500 iterations



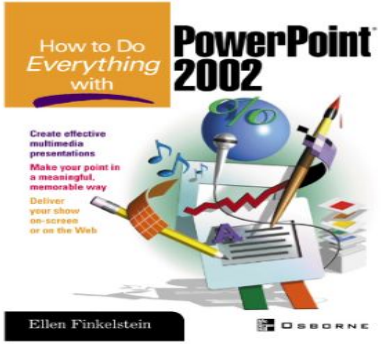
2500 iterations



8000 iterations

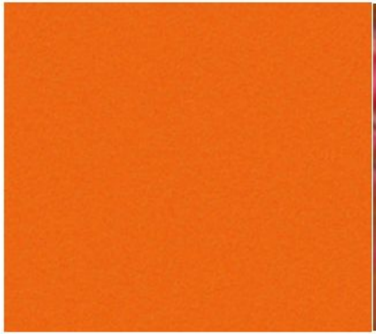


Corrupted Image



Original Image

Region Inpainting Results



Random Noise

500 Iterations

1500 Iterations

2500 Iterations

8000 Iterations

Corrupted Image

Original Image

Region Inpainting Results

0 Iterations

500 Iterations

1500 Iterations

4000 Iterations

8000 iterations

Corrupted Image

Original Image

Contributions

Aman Khandelwal : Prepared the code for Image Conversion (i.e. image pixels change, adding uniform noise, region removal, removing random pixel). Appling Hourglass model for denoising the images and training on noisy images. Showing the restored image and loss for every 500 iterations. Plotting the graph for loss vs epoch. Also evaluated and showing the loss comparison plot for reconstruction task using original Image vs Image+Noise.

Nikhil Khemchandani : Made Standard Hourglass Model with upsampling , downsampling and skip layers using tensorflow-keras and used Adam optimizer for optimizing weights with subject to MSE Loss function . Trained Region Inpainting on masked Images indicating the Missing Regions and showed model accurately restored regional pixels,demonstrating the potential of this approach for image restoration tasks.

Piyush Singh : Text Inpainting and Randomly Damaged Pixels Inpainting. Both were trained with corresponding masks indicating the missing or damaged areas. Results shows that the model accurately restored missing text and damaged pixels, demonstrating the potential of this approach for image restoration tasks.

Results and Code Link

Implementation Link:
https://drive.google.com/drive/folders/14EWBmGDl0wENGPBc9TouN8tNK-K-CEAz?usp=sharing

Result Link:
https://drive.google.com/drive/folders/15IuyqH3MXX3cSMynFcA4rlhAF9T8YFEB?usp=sharing


Thank
You