DEEP IMAGE PRIOR

PRESENTED BY-

1. MANOJ GEMBALI (A20527288)
2. TUSHAR GWAL (A20449419)

Problem Statement and Background

- **Image Restoration Challenges :**

  Denoising: Remove unwanted noise (random variations, sensor noise) while preserving image details

  Inpainting:
  - Text removal: Eliminate overlaid text
  - Hole filling: Fill missing or damaged regions

  Large-Scale Reconstruction: Handle cases with significant image corruption like large missing regions or severe degradation

- **Restoring image using Traditional Approach:**

  i. **Prior in Image Restoration:** In image processing, priors encode our assumptions about what makes an image look "natural." When restoring degraded images (noisy, blurry, or incomplete), priors help:
  - Distinguish between valid image content and noise/artifacts
  - Guide reconstruction when information is missing
  - Ensure the restored image looks realistic

  ii. **CNN** are trained on complex image priors from large datasets which provides them:
  - End to End learning where CNNs can learn to map corrupted images directly to clean images.
  - Hierarchical feature learning which allows Deep networks to capture both low-level and high-level image features.

Problem Statement and Background

Limitations of conventional approaches:

- Reliance on large datasets
- Poor generalization to unseen degradation types

Deep Image Prior (DIP) as a novel solution:

- Uses CNN architecture as an implicit prior
- No need for external training data

Proposed Solution - Deep Image Prior

In traditional approach, When dealing with degraded images , we need two key components:

- **Data Term**: Measures how well our solution matches the degraded input
- **Prior Term**: Guides the reconstruction toward natural-looking images

Traditional reconstruction can be formulated as:

$$x^* = \underset{x}{\operatorname{argmin}}\, E(x; x_0) + R(x)$$

- where:
- E(x; x0) is the data term
- R(x) is the prior/regularizer
- x0 is the degraded image
- x* is the restored image

Proposed Solution - Deep Image Prior

- Whereas in Deep Image Prior (DIP), as opposed to traditional image restoration tasks with explicit priors, the approach leverages the structure of a convolutional neural network itself as an implicit prior for natural images. Instead of using pre-trained networks or explicitly defined regularization terms, DIP employs an untrained CNN that maps fixed random noise to an output image. The network weights become the optimized parameters, effectively reparametrizing the image as

  $x = f_\theta(z)$, where f_θ is the untrained CNN, z is fixed random noise, and θ are the network parameters.

- The optimization objective is to find the optimal network parameters θ* that minimize the error between the network output and the corrupted image, without any explicit prior term.

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

where:

f_θ is an untrained CNN z is fixed random noise

θ are network parameters

No explicit prior term R(x) Prior is implicit in CNN architecture

Why it works?

Key Insight from CNN Behavior:

- Critical Observation: CNNs can fit random labels in classification
  - Shows networks have massive capacity to overfit
  - Yet they still exhibit strong inductive bias towards structured data
- Important implication: Architecture itself imposes meaningful structure

Network's "Natural Impedance":

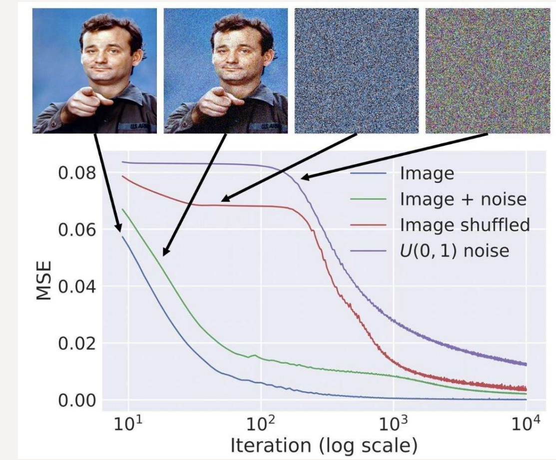
- Fast convergence for natural image structures
- Slow convergence for noise/artifacts
- Network architecture naturally resists noise while preserving image content
- Early stopping exploits difference in fitting speeds

Multi-scale Structure Properties:

- Generates and organizes features hierarchically from random noise
- Gradually transforms noise into meaningful image content
- Skip connections preserve and refine details across scales
- Network structure develops and exploits self-similarities during optimization

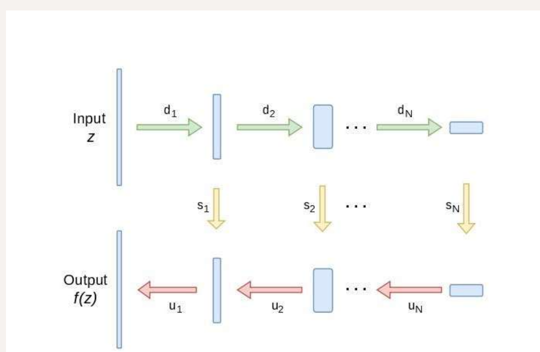
Implicit Regularization:

- Network structure biases towards smooth, structured solutions
- Convolutional layers promote natural image statistics



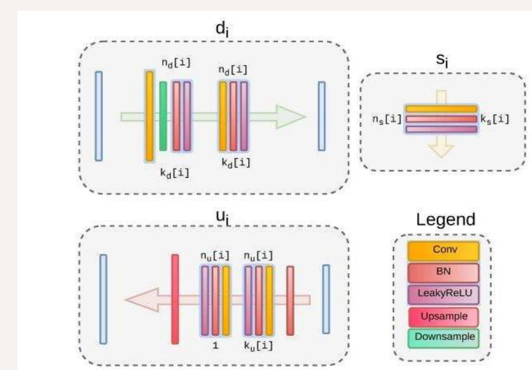
Deep Image Prior Architecture

- DIP is **U-Net-Inspired** Encoder-Decoder Structure
- **Encoder:**
  - Processes and gradually transforms random noise into structured representations
  - Depth affects the complexity of patterns that can emerge
- **Decoder:**
  - Transforms encoded representations into image-like output
  - Combines high-level features with spatial information from the encoder.



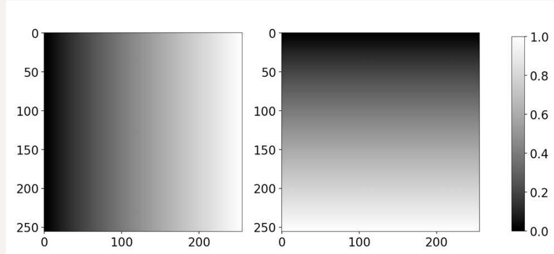
Deep Image Prior Architecture

- **Skip Connections:**
  - Links corresponding layers in the encoder and decoder.
  - Helps retain important details that might be lost during encoding.
- **Reflection Padding:**
  - Pads the image by reflecting its borders.
  - Minimizes edge artifacts and ensures smoother image edges.



Implementation Details

- Input Types:
  - Random Noise (Denoising, Inpainting, Reconstruction):
    - Utilized as the initial input to the network.
  - Meshgrid (Hole Filling):
    - Encodes spatial information explicitly.
    - Advantageous for reconstructing missing regions with spatial coherence.



- Skip Connections:
  - Implemented in Denoising, Inpainting, Reconstruction:
    - **Purpose:** Preserve spatial details by transferring high-resolution features from encoder to decoder.
  - None in Hole Filling:
    - **Reason:** Focus on reconstructing large missing regions where skip connections may introduce artifacts or impede the generation of coherent structures.
- Upsampling Methods:
  - Bilinear Upsampling (Denoising & Reconstruction):
    - Provides smooth and continuous upsampling.
    - Enhances the quality of reconstructed images by maintaining gradient continuity.
  - Nearest Neighbor Upsampling (Inpainting & Hole Filling):
    - Preserves hard edges and sharp transitions, which are beneficial for filling discrete or well-defined missing regions.

Training Process

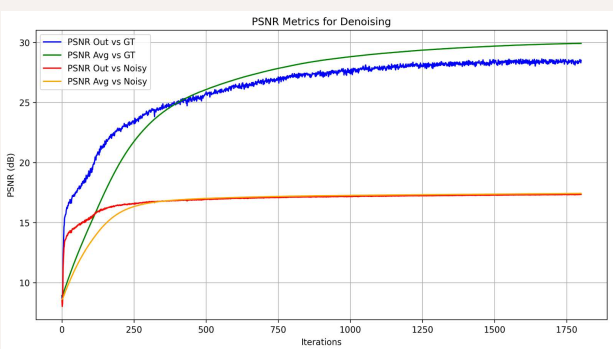
- Adam Optimizer
  - Adaptive learning rates
  - Efficient gradient handling
- Loss Function
  - Mean Squared Error (MSE) for All Tasks
  - Masked MSE for Inpainting, Hole Filling, Reconstruction
    - Calculated only over known regions
    - Focuses learning, preserves unaltered areas
- Noise-Based Regularization
  - Adds Gaussian noise to input during training
  - Improves generalization and exploration
  - Prevents overfitting to corrupted input
- Checkpoint Mechanism
  - Tracks Peak Signal-to-Noise Ratio (PSNR)
  - Saves best-performing models
  - Prevents loss due to overfitting or fluctuations

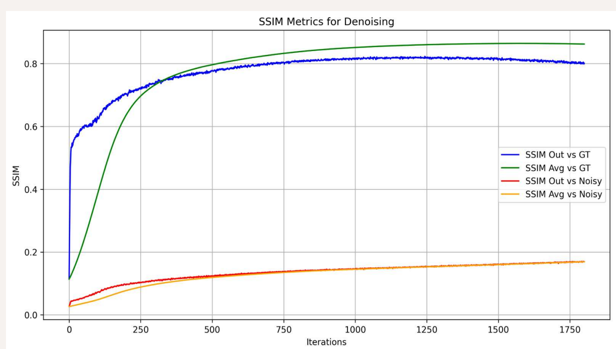
Quantitative Results - Denoising

- **PSNR (Peak Signal-to-Noise Ratio):**
  Measures image quality based on pixel-level differences.
  Higher values (typically above 30 dB) indicate better quality.
- **SSIM (Structural Similarity Index Measure):**
  Assesses image quality based on structural similarities.
  Values closer to 1 indicate better quality, with >0.95 considered excellent.

Metrics

	PSNR	SSIM
Final_vs_GT	26.633632738881374	0.8262966
Avg_vs_GT	29.922273164622798	0.8624169
Final_vs_Noisy	17.159526820560004	0.175373
Avg_vs_Noisy	17.42376855052601	0.16904747



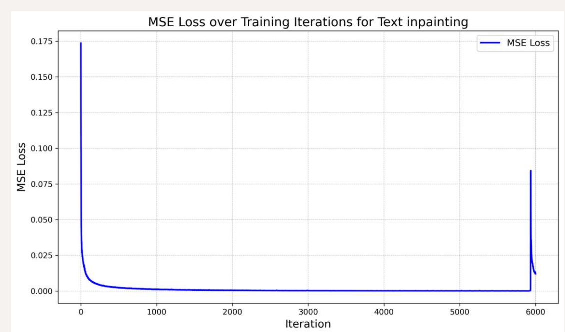


Quantitative Results – Inpainting

- The SSIM metric is used to evaluate how closely the filled-in regions resemble the ground truth.
- High SSIM scores indicated successful reconstruction in masked regions.

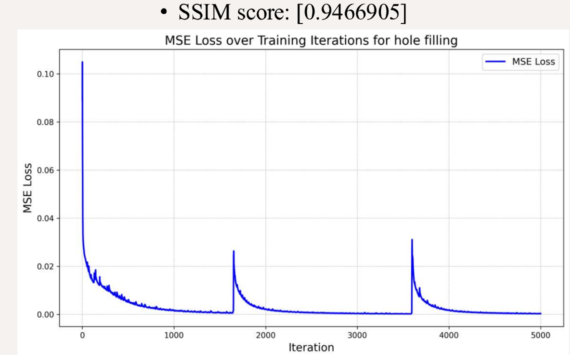
Text Inpainting

- SSIM score: [0.96201557]



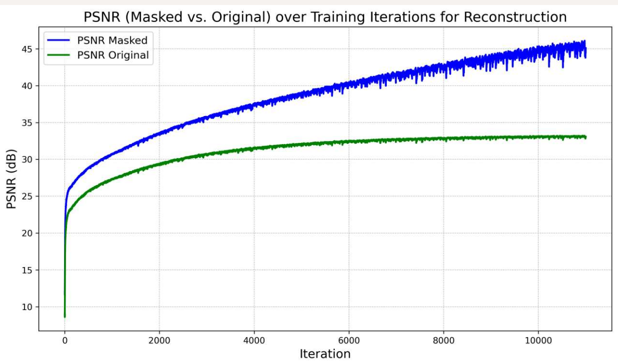
Hole filling

- SSIM score: [0.9466905]

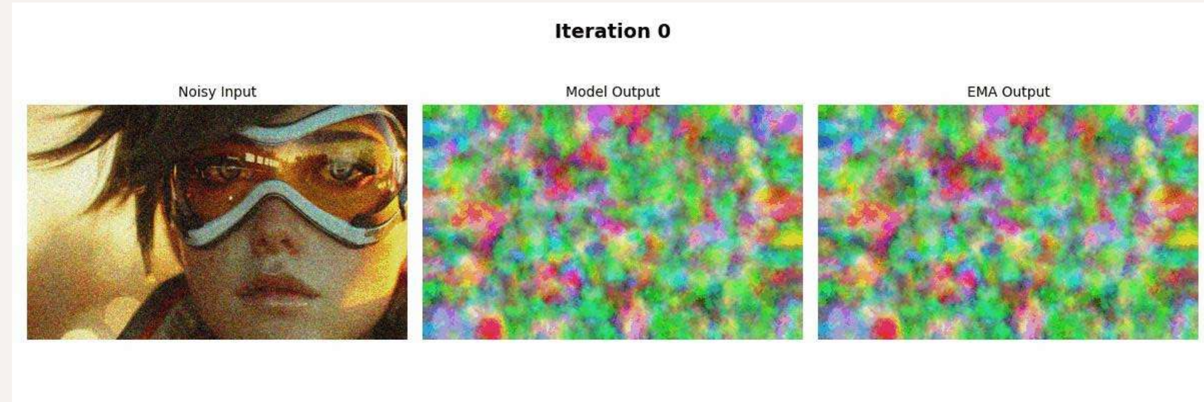


Quantitative Result – Image Reconstruction

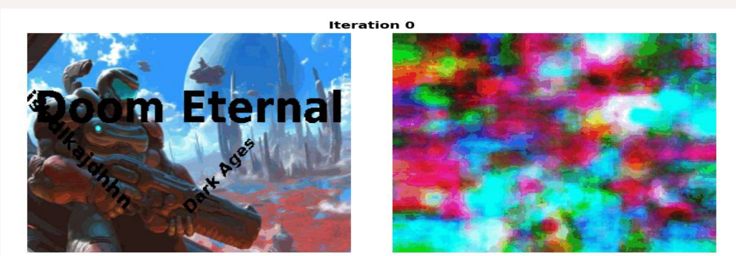
- SSIM score: [0.939079]



Model Demonstration - Denoising



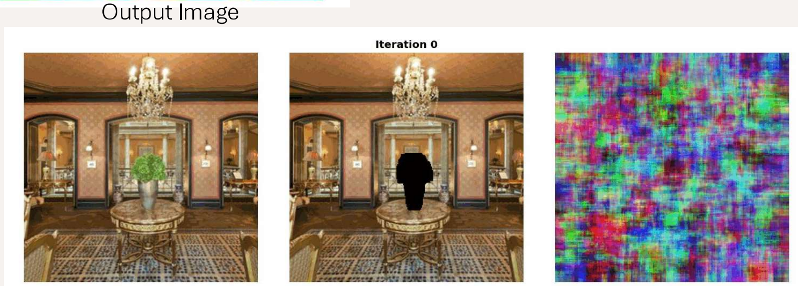
Model Demonstration - Inpainting



Corrupted Image Output Image

Text Inpainting

Hole Filling



Original Image Corrupted Image Output Image

Model Demonstration – Reconstruction

Original Image Corrupted Image Output Image

Conclusion and Future Work

- **Strengths of Deep Image Prior:**
  - Effectiveness for single-image tasks
  - No need for external training data
- **Limitations:**
  - Computational intensity
  - Risk of overfitting
- **Potential improvements:**
  - Adaptive early stopping mechanisms
  - Exploring alternative architectures

Thank You

Any questions ?