Image Denoising with DnCNN

Saurabh Jondhale

1 Introduction

Image noise is like "visual static" that makes photos look grainy. We use **DnCNN** (a deep learning model) to remove noise.

1.1 Key Idea

Instead of guessing the clean image directly, DnCNN **predicts the noise** and subtracts it:

$$\text{Clean Image} = \text{Noisy Image} - \text{Predicted Noise}$$

2 How DnCNN Works

2.1 Step 1: Add Synthetic Noise

We corrupt clean images with artificial noise (e.g., Gaussian noise):

```
noisy_image = clean_image + random_noise
```

2.2 Step 2: Model Architecture

DnCNN has:

- **Convolutional Layers** (3×3 filters)
- **Batch Normalization** (stabilizes learning)
- **Residual Learning** (predicts noise, not the image)

3 Math Simplified

3.1 Training Goal

Minimize the difference between **true noise** and **predicted noise**:

$$\text{Loss} = \|(\text{Noisy} - \text{Clean}) - \text{Model}(\text{Noisy})\|^2$$

3.2 Example

Clean Pixel	Noise	Noisy Pixel	Predicted Noise	Denoised Pixel
100	+10	110	+9	101
150	-20	130	-18	148

Table 1: Noise removal example

4 Code Implementation

4.1 PyTorch Model

```
class DnCNN(nn.Module):
    def __init__(self):
        super().__init__()
        self.layers = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=3, padding=1),
            nn.ReLU(),
            nn.BatchNorm2d(64),
            # ... more layers ...
        )
    def forward(self, x):
        return x - self.layers(x)  # Subtract predicted noise
```

4.2 Training Loop

```
for epoch in range(50):
    loss = criterion(model(noisy), noisy - clean)
    loss.backward()
```

Metrics:

- PSNR: 28.5 dB $\rightarrow$ 32.1 dB
- SSIM: 0.85 $\rightarrow$ 0.93

5 References

1. Zhang, K. et al. (2017). *"Beyond a Gaussian Denoiser: Residual Learning for Image Denoising."* arXiv:1608.03981.

2. PyTorch Documentation. *"Convolutional Neural Networks."*

Compiled By: Saurabh Jondhale
Contact: jondhalesaurabh27@gmail.com
GitHub: https://github.com/saurabhjondhale/Denoise