Dual-Domain Deep Learning for Accelerated MRI Reconstruction

Samanyu Satheesh
Nikhil Kuniyil
Megna Srinivas

Motivations

MRI provides excellent soft tissue contrast, but is slow due to full k-space acquisition. Undersampling reduces time but introduces artifacts/errors

Solution: Use neural networks

Acceleration/Undersampling Challenge
- Acquiring a subset of k-space lines
- Massive reduction in scan time, but loss of some k-space data

How to recover missing information?
- Deep Learning
- Neural networks can be trained to learn optimal reconstruction
- NNs can learn patterns and relationships automatically
- Two fundamental approaches: learn in image or k-space domain?

Input
Undersampled MRI data

Output
High-quality reconstructed MRI image

Challenge
Must correct aliasing without hallucinating anatomy

Dataset – fastMRI Knee MRI (NYU)

- **Dataset source**
  - fastMRI Knee Dataset (NYU, 2018 benchmark)
- **Content**
  - Raw k-space measurements
  - Fully sampled ground-truth images
  - Proton-density (PD) and PD-fat-suppressed (PD-FS) contrasts
- **Preprocessing**
  - Complex-valued IFFT $\rightarrow$ 2-channel (real/imag)
  - Per-slice normalization
- **Train/val/test Split**
  - Train: 2,684 slices
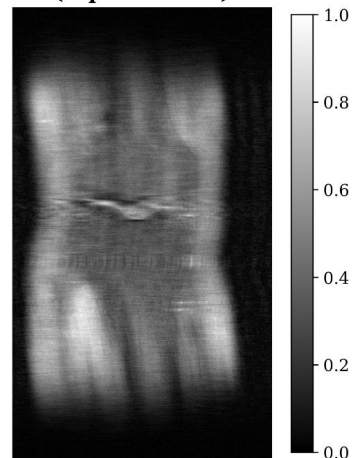 - Val: 604 slices
  - Test: 615 slices

An example slice from FastMRI dataset

Undersampling & Zero-Filled Reconstruction

- **Undersampling**
  - Apply cartesian mask to raw k-space
  - Reduces scan time (R = 4)
  - Introduces aliasing artifacts
- **Zero-Filled Reconstruction**
  - Inverse FFT of undersampled k-space
  - Contains basic anatomy + strong artifacts

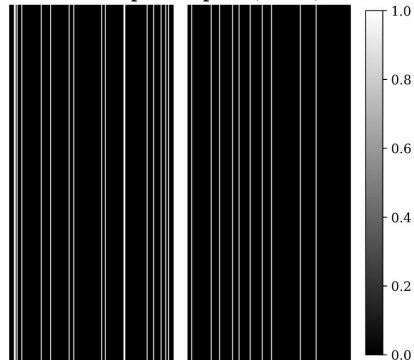
Zero-Filled Reconstruction (Input to CNN)

Example from dataset.py

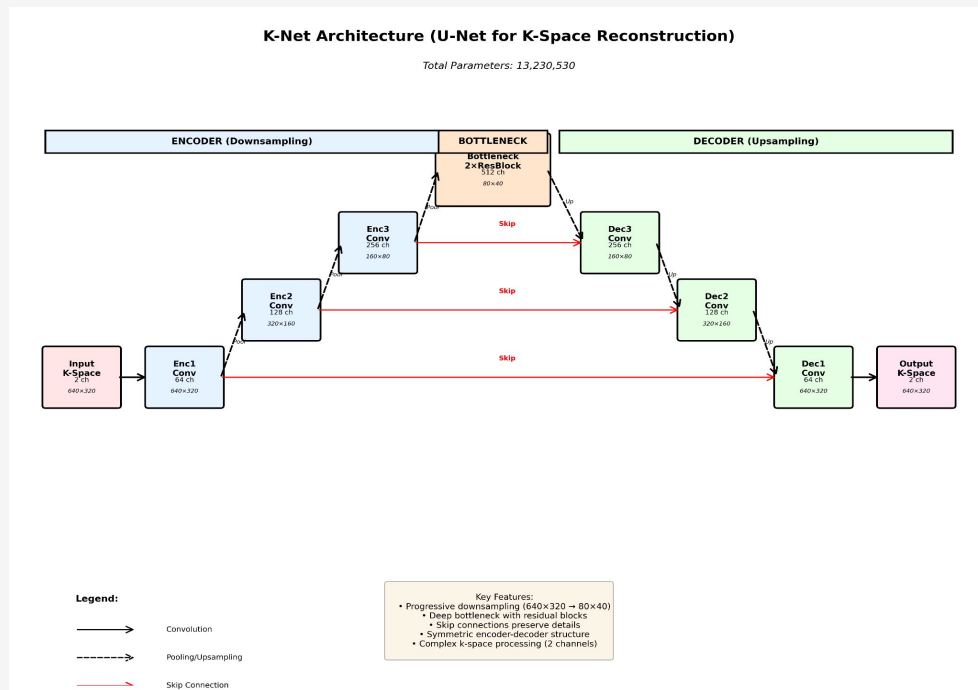Fully Sampled Image | Undersampled k-space (R=8.8)

Model 1: K-Space U-Net

- **Input**: 2-channel k-space with missing lines
- **Output**: 2-channel complete k-space
- **Goal**: understand frequency structure from a physics-grounded perspective
- **Features**:
  - Low-level features (early layers): recognizing frequency transitions and missing data, understanding symmetry of k-space
  - Mid-level features (middle layers): k-space smoothness, cross-frequency relationships
  - High-level features (bottleneck): solidifying Fourier properties, global frequency structure
- **Training**: 20 epochs (~6 hours of training), Adam optimizer

Model 1: K-Space U-Net

- **U-Net Structure**
- Captures features through convolution and pooling into small, feature-rich representations
- Upsampling then expands it back to full resolution
- Utilizes skip connections to preserve fine details

K-Net Architecture (U-Net for K-Space Reconstruction)

Total Parameters: 13,230,530

From visualize_architecture.py

Model 1: Results
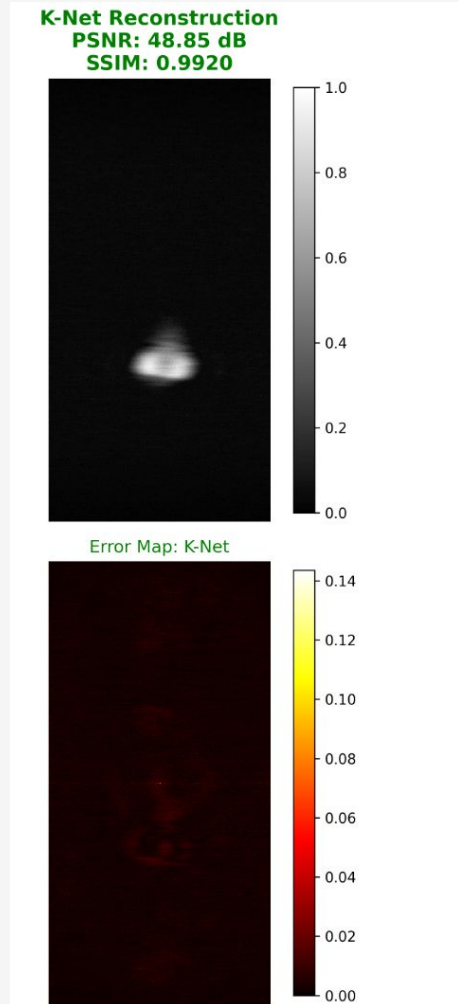
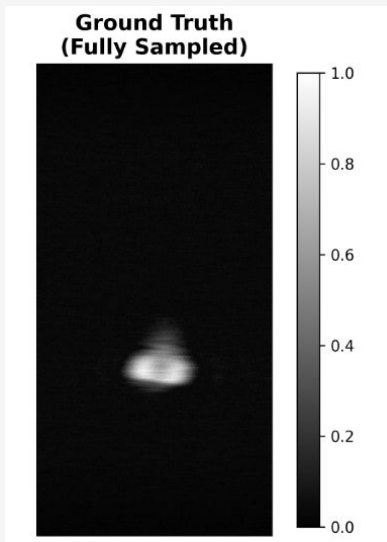- **Training Results (20 epochs)**
  - PSNR: 44.55 ± 1.53 dB
  - SSIM: 0.9832 ± 0.0082
  - NMSE: 0.001121 ± 0.000296
- **Test Set Results**
  - PSNR: 42.89 ± 1.53 dB
  - SSIM: 0.9797 ± 0.0082
  - NMSE: 0.001090 ± 0.000296

- **Comments:**
  - Only a 1.7dB drop in test, suggests minimal overfitting and good generalization
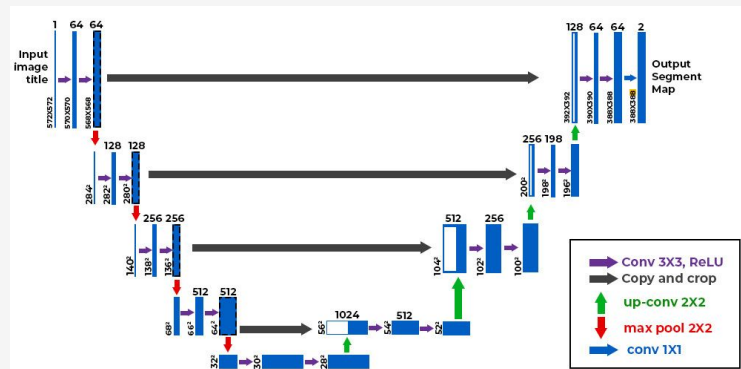 - Low standard deviation tells us the model is consistent

6

Ground Truth (Fully Sampled)

K-Net Reconstruction
PSNR: 48.85 dB
SSIM: 0.9920

Error Map: K-Net

From visualize_knet.py

Model 2: Image-Domain U-Net

- **Input:** Zero-filled reconstruction from undersampled k-space
- **Architecture:** encoder-decoder CNN with skip connections
- **Goal:** remove aliasing + recover fine anatomical details
- **Complex handling:** 2-channel real/imag or magnitude representation
- **Residual learning:** network predicts correction to the input image
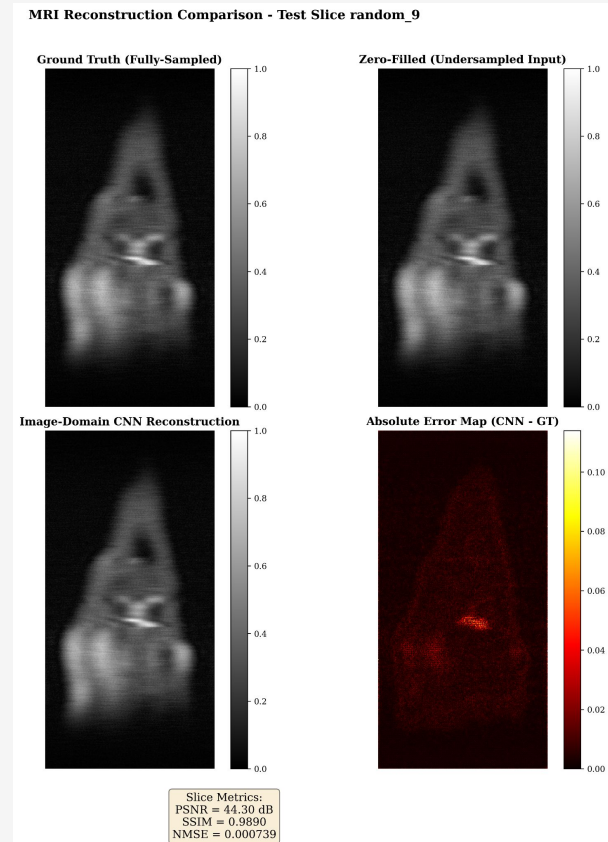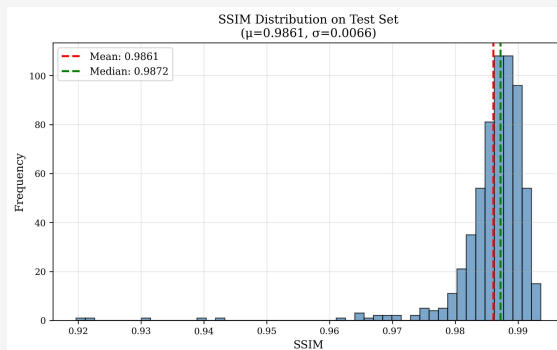- **Training:** supervised with MSE loss on ground-truth images
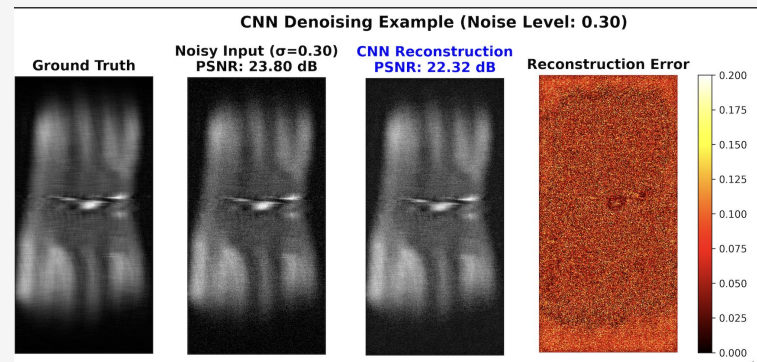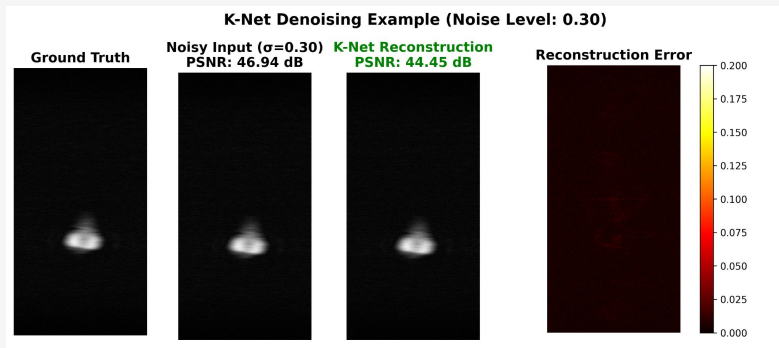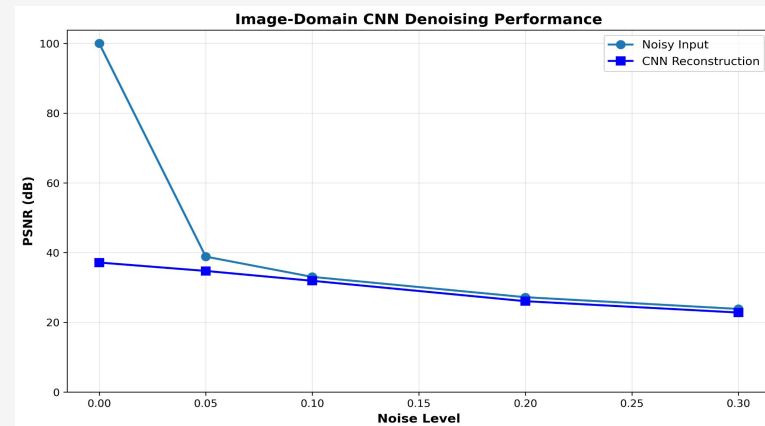

Specified in model.py

Reconstruction Results (Image-Domain U-Net)

- **Training Results (20 epochs)**
  - PSNR: 40.27 dB
  - SSIM: 0.9853
  - NMSE: 0.002397
- **Test Set Results**
  - PSNR: 41.52 ± 3.28 dB
  - SSIM: 0.9861 ± 0.0066
  - NMSE: 0.002048 ± 0.003183

From metrics.py

Denoising Performance

K-Net Denoising Performance

Image-Domain CNN Denoising Performance

K-Net Denoising Example (Noise Level: 0.30)

Ground Truth | Noisy Input (σ=0.30) PSNR: 46.94 dB | K-Net Reconstruction PSNR: 44.45 dB | Reconstruction Error

CNN Denoising Example (Noise Level: 0.30)

Ground Truth | Noisy Input (σ=0.30) PSNR: 23.80 dB | CNN Reconstruction PSNR: 22.32 dB | Reconstruction Error

9

From test_knet_noise.py

Overview

Aspect	K-Space U-Net	Image U-Net
Input	2-channel k-space	Zero-filled image
Output	Completed k-space	Corrected image
Strength	Physics-grounded; learns frequency structure	Learns anatomy; excellent at reducing visible artifacts
Features learned	Mathematical relationships, symmetry, Fourier behavior	Edges, textures, anatomical priors
Risk	Harder to train; sensitive to noise	May hallucinate features if poorly trained
Complexity	High (complex-valued CNN)	Medium (standard CNN)
Training time	~6 hours for 20 epochs	~7.5 hour for 20 epochs
Common literature	KIKI-Net, MoDL, VNets	DeepCascade, U-Net, AUTOMAP

Proposed Method: Dual-Domain Network

K-space CNN (K-Net)	Data consistency layer	IFFT/ U-Net	Output

Constraints:
- Dual-domain networks = significantly heavier
- Require multiple FFT/IFFT layers inside training loop
- Training time ~12–24 hours on a typical laptop GPU
- Our dataset size had to be reduced (≤10 fastMRI files)

We prioritized:
- Implementing both single-domain pipelines
- Running reproducible experiments
- Achieving high-quality reconstructions within the time limit

Results:
- We trained k-space U-Net and image U-Net separately
- Compared behaviors, strengths, weaknesses
- Outlined how they would be combined for a final model

Literature & Citations

• Zhu et al., AUTOMAP (Nature 2018): domain transform learning. PubMed
• Eo et al., KIKI-Net (MRM 2018): cross-domain CNNs alternating k-space and image
CNNs. PubMed
• Zhou & S. K. Zhou, DuDoRNet (CVPR 2020): dual-domain recurrent network. CVF
Open Access
• Aggarwal et al., MoDL (2018): model-based deep learning with data consistency.
PMC
• Yaman et al., SSDU / self-supervised physics-guided learning (2020): training
without
fully sampled data.
• fastMRI dataset paper & site (Zbontar et al., 2018; fastMRI site): benchmarking &
data.