Pixel-Wise Deep Reinforcement Learning Approach for Ultrasound Image Denoising

PROJECT REPORT
REINFORCEMENT LEARNING

Submitted by:

K. BALA SAI MANVITHA (CS22B1030)
M. KRISHNA TEJA (CS22B1044)
K. PRANAY SAI (CS22B1054)
V. SRI MANASWINI (CS22B2030)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING, KANCHEEPURAM

May 11, 2025

Contents

Abstract

Ultrasound imaging is widely used in clinical diagnostics due to its non-invasive and real-time imaging capabilities. However, one of the major limitations of ultrasound images is the presence of speckle noise, which significantly degrades image quality and interpretability. Traditional denoising methods often rely on handcrafted filters or deep convolutional networks, which are typically non-transparent in terms of decision-making. In this project, we propose a novel denoising technique using Pixel-Wise Deep Reinforcement Learning (PixelRL), where each pixel is treated as an agent capable of taking interpretable filtering actions. The model is built on an A3C-based policy and value network with recurrent memory and is evaluated on the BSD68 dataset. Experimental results show significant improvement in PSNR and visual quality.

1 Introduction

Ultrasound imaging is an essential diagnostic tool used in various medical fields such as cardiology, obstetrics, and oncology. Despite its advantages in terms of safety and cost-effectiveness, ultrasound imaging suffers from inherent noise known as speckle noise. Speckle appears due to interference from multiple scatterers and reduces both contrast and resolution, making it challenging to interpret anatomical structures.

While traditional approaches use deterministic filtering (e.g., Gaussian, median, bilateral), they often fail to adapt to local image features. Deep learning approaches have improved denoising quality, but their black-box nature limits clinical trust and interpretability. Reinforcement learning (RL), particularly at the pixel level, introduces a paradigm shift by allowing adaptive, interpretable, and localized image enhancement.

This report introduces a PixelRL model, wherein each pixel acts as an agent, choosing from a set of predefined filtering actions based on its local context. The training is carried out using A3C (Asynchronous Advantage Actor-Critic), and the results demonstrate significant improvement in both PSNR and qualitative quality.

2 Related Work

2.1 Traditional Denoising Methods

Conventional denoising techniques include Gaussian smoothing, median filtering, bilateral filtering, and anisotropic diffusion. These methods aim to suppress noise while preserving edges and structures. Gaussian filters apply a weighted average that reduces high-frequency components, but they often blur fine structures. Median filters are edge-preserving but struggle with high levels of noise. Bilateral filters preserve edges better by considering both spatial and intensity differences. However, all these methods are non-adaptive and apply the same operation to all regions, regardless of content, which limits their effectiveness.

2.2 CNN-Based Denoising

Deep learning has become a popular approach for denoising. Models such as DnCNN and U-Net have shown high performance on natural image denoising tasks. These networks learn mappings from noisy to clean images through supervised learning on large datasets. While they achieve high PSNR scores, they lack interpretability and require extensive data and compute. Moreover, these models tend to overfit to specific noise distributions and do not generalize well without retraining.

2.3 Pixel-Wise Reinforcement Learning

The PixelRL approach was first introduced to tackle the problem of image denoising in an interpretable and adaptive manner. By treating each pixel as an independent agent capable of choosing actions based on its local context, PixelRL can adapt to different noise patterns and preserve image details more effectively. Jarosik et al. demonstrated that reinforcement learning could achieve competitive denoising results while providing a visual map of operations performed across the image.

3 Proposed Method

3.1 Dataset and Preprocessing

We use the BSD68 dataset, a standard benchmark for image denoising. The dataset comprises 68 grayscale images with diverse textures and structures. To simulate realistic noise conditions, we add Gaussian noise with a standard deviation of $\sigma = 15$. The noisy images are then normalized to a 0–1 range. Each image is cropped into smaller patches of size 70x70 to facilitate mini-batch training. During training, each patch undergoes random augmentation such as horizontal flipping and rotation to increase data diversity and reduce overfitting.

3.2 Action Space

In our framework, each pixel-agent can select one of nine actions. These actions are designed to mimic basic image processing operations and provide interpretability:

1. Increase pixel intensity by a fixed value
2. Decrease pixel intensity by a fixed value
3. Apply a Gaussian filter with small variance
4. Apply a Gaussian filter with large variance
5. Apply a bilateral filter with weak smoothing
6. Apply a bilateral filter with strong smoothing
7. Apply a median filter
8. Apply a box filter
9. Do nothing (neutral action)

This discrete action space enables selective and localized changes to the image and encourages the network to learn how different filters affect local noise patterns.

3.3 Network Architecture: MyFcn

The proposed network architecture, MyFcn, is a fully convolutional network tailored for pixel-wise RL. It consists of:

Input Processing: An initial convolution layer transforms the grayscale input image into 64-channel feature maps. This is followed by a sequence of dilated convolutions with varying dilation rates to capture local and global context efficiently.

ConvGRU Module: Inspired by recurrent neural networks, we implement a ConvGRU-like mechanism that maintains a spatial hidden state. This enables the network to retain memory of previous actions and image states, facilitating temporal learning across multiple steps.

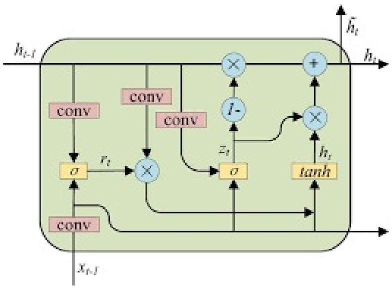

Figure 1: ConvGRU

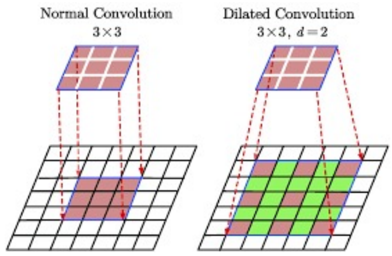

Figure 2: Dilated Conv

Policy and Value Heads: The shared features are fed into two separate branches:

- Policy branch outputs a probability distribution over the 9 actions for each pixel.
- The value branch estimates the expected future reward for each state of pixels using PSNR value.

This dual-head architecture enables stable and efficient learning using the actor-critic method.

3.4 Learning Framework

The training framework is based on the A3C (Asynchronous Advantage Actor-Critic) algorithm. Each episode consists of a sequence of 5 steps where the agent interacts with the image environment by selecting actions and receiving feedback. The rewards are calculated based on the improvement in PSNR (Peak Signal-to-Noise Ratio) or reduction in MSE (Mean Squared Error) between steps. The total loss is a weighted sum of:

1. Policy loss: Encourages selection of actions that lead to higher rewards
2. Value loss: Reduces error in value function prediction
3. Entropy loss: Promotes exploration by penalizing overly confident predictions

The model is trained for 50,000 episodes using the Adam optimizer with learning rate decay.

4 Implementation Details

The entire system is implemented in Python using TensorFlow 2.x. Image handling and preprocessing are done using OpenCV and NumPy. The training and testing processes are organized into modular scripts:

- **mini_batch_loader:** Loads and augments patches from BSD68
- **MyFcn:** Defines the neural network architecture

- **State:** Custom environment applying pixel-wise actions
- **pixelwise_a3:** Implements A3C learning logic with policy/value updates

Training is conducted with a batch size of 64, and evaluation is performed every 3000 episodes. The model checkpoints and results are stored for analysis.

5 Testing and Evaluation

To evaluate the performance, the trained model is tested on unseen patches from the BSD68 dataset. Each patch is corrupted with Gaussian noise and passed through the trained model for five action steps. At each step, actions are chosen based on the current policy, and the resulting image is updated.

Figure 3: Original Image

Figure 4: Denoised Image

Figure 5: Original Image

Figure 6: Denoised Image

Figure 7: Original Image

Figure 8: Denoised Image

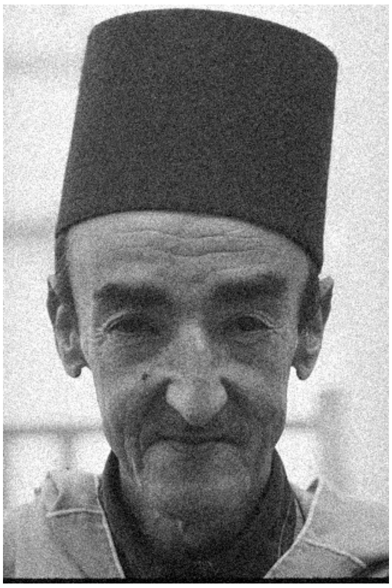

Figure 9: Original Image

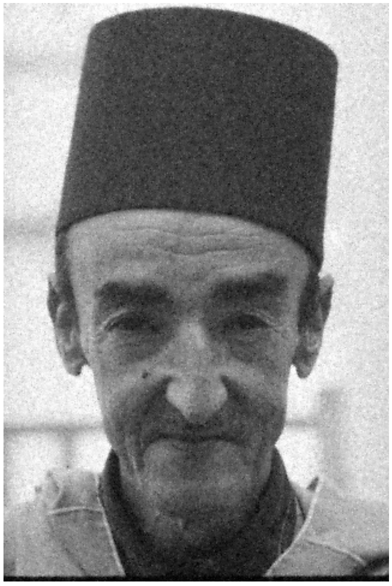

Figure 10: Denoised Image

Evaluation metrics include:

- **PSNR (Peak Signal-to-Noise Ratio):** Measures reconstruction quality
- **Qualitative Outputs:** Noisy, clean, and denoised images are saved for visual comparison
- **Total Reward:** Summation of PSNR-based rewards across the episode

Results indicate a consistent improvement in PSNR and enhanced visual quality with reduced speckle and sharper edges.

6 Results

Experimental results show that the model achieves strong denoising performance. On average, the denoised images exhibit:

- **Average Input PSNR:** 24.80 dB

- **Average Output PSNR:** 28.54 dB
- **Gain:** +3.74 dB improvement

Visual inspection confirms that the model effectively suppresses noise while preserving important anatomical structures. Action maps demonstrate the interpretability of the approach, showing which filters were applied in different regions.

7 Conclusion

In this project, we successfully implemented a Pixel-Wise Deep Reinforcement Learning model for ultrasound image denoising. Our approach demonstrates that reinforcement learning can offer interpretable and adaptive solutions for pixel-level noise reduction. The integration of filtering actions and temporal memory makes this method robust to varying noise patterns and provides transparency often lacking in CNN-based models.

Future directions include applying the framework to real ultrasound datasets, incorporating more sophisticated actions (e.g., wavelet transforms), and experimenting with advanced RL algorithms like PPO or SAC to improve convergence and stability.

8 Individual Contributions

- **K. Bala Sai Manvitha (CS22B1030)**
  - Conducted the literature review on denoising techniques and compiled insights to guide model development.
  - Contributed to the policy and value network tuning, and coordinated the writing of the final report and presentation.
- **M. Krishna Teja (CS22B1044)**
  - Handled dataset preparation, noise modeling, and preprocessing pipeline.
  - Assisted in implementing and validating the evaluation metrics (PSNR, qualitative comparisons).
- **K. Pranay Sai (CS22B1054)**
  - Implemented the neural network architecture (MyFcn) including ConvGRU and dual policy-value heads.
  - Led the integration and training using the A3C reinforcement learning framework.
- **V. Sri Manaswini (CS22B2030)**
  - Developed the custom image environment and pixel-agent action simulation logic.
  - Coordinated testing, visualization of outputs, and final presentation formatting.

9 References

- Piotr Jarosik et al., “Pixel-Wise Deep Reinforcement Learning Approach for Ultrasound Image Denoising,” IEEE Transactions on Medical Imaging, 2020.
- Zhang et al., ”Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising,” IEEE Transactions on Image Processing, 2017.
- Mnih et al., ”Asynchronous Methods for Deep Reinforcement Learning,” ICML 2016.