Binary Mask Denoising: Detailed Methodology, Implementation, and Analysis

Mohamed Khalil Kiri

October 26, 2025

1 Problem Statement

The main objective of this exercise was to reconstruct clean binary masks from noisy grayscale images. Each input image $x \in R^{H \times W}$ contained a faint binary pattern representing a segmentation mask, overlaid with stochastic noise. Formally, the observed input is $x = m + n$, where m is the underlying clean mask and n represents additive noise. The goal is to learn a mapping function $f_\theta(x)$, parameterized by a neural network, that outputs an estimate $\hat{m}$ satisfying $\hat{m} \approx m$.

The task presents several inherent challenges:

- The noise is non-trivial, composed of Gaussian noise, salt-and-pepper corruption, and occasional blur applied randomly across the dataset.
- Masks contain multiple geometric shapes, such as circles, rectangles, and polygons, which require the network to capture spatial relationships across the image.
- The mapping from noisy inputs to clean masks is fundamentally ambiguous: different noisy realizations can correspond to the same underlying mask.

2 Methodology

2.1 Synthetic Dataset Generation

Given the lack of a real dataset, a synthetic dataset was constructed to emulate realistic segmentation challenges while controlling the complexity.

Mask Generation Each mask consists of 2 to 4 randomly positioned geometric shapes:

- **Circles**: defined by a center and radius.
- **Rectangles**: defined by the top-left corner, width, and height.
- **Polygons**: approximated with random vertices forming simple polygons.

This setup ensures diversity in shapes and encourages the network to generalize to different geometries.

Noise Modeling To simulate realistic acquisition noise:

- Gaussian noise with varying standard deviation to emulate sensor inconsistencies.
- Salt-and-pepper noise applied randomly to a small fraction of pixels.
- Gaussian blur applied randomly to some samples, introducing partial occlusion of edges.

Data Augmentation To improve generalization and increase sample diversity without storing additional images, we applied on-the-fly augmentations:

- Horizontal and vertical flips.
- Rotations of 90°, 180°, or 270°.

These transformations ensure the model encounters multiple variations of the same underlying patterns, which is crucial for robustness.

2.2 Model Architecture

Given the spatially-structured nature of the task, a UNet-inspired convolutional neural network was selected. The main design principles are:

- **Encoder-Decoder Structure**: The encoder extracts hierarchical features from the input, progressively reducing spatial dimensions, while the decoder upsamples features to reconstruct the mask at the original resolution.
- **Skip Connections**: Direct connections between corresponding encoder and decoder layers allow high-resolution details to bypass the bottleneck, preserving fine-grained spatial information essential for accurate mask reconstruction.
- **Convolutional Blocks**: Each block contains two consecutive 3x3 convolutional layers with batch normalization and ReLU activation. Dropout is introduced in deeper layers to reduce overfitting risk.
- **Output Layer**: A 1x1 convolution followed by a sigmoid activation produces per-pixel probabilities, effectively generating a soft mask.

Rationale UNet architectures are widely adopted in image segmentation tasks, particularly in situations with limited data. Their ability to retain spatial fidelity through skip connections and capture context at multiple scales makes them well-suited for reconstructing noisy binary masks.

2.3 Loss Function

To simultaneously enforce pixel-wise accuracy and region-level overlap, a composite loss function was employed:

$$\mathcal{L} = \alpha \cdot \text{BCE} + (1 - \alpha) \cdot \text{DiceLoss}, \quad \alpha = 0.7$$

Binary Cross-Entropy (BCE) BCE penalizes incorrect pixel predictions individually, ensuring that each pixel is classified correctly.

Dice Loss Defined as:

$$\text{Dice} = \frac{2\sum_i p_i t_i}{\sum_i p_i + \sum_i t_i + \epsilon},$$

where p_i and t_i are predicted and ground truth pixels. Dice loss emphasizes spatial overlap and stabilizes training when the mask occupies a small fraction of the image.

2.4 Training Pipeline

The training procedure follows standard best practices for image-to-image regression:

- **Optimizer**: Adam, chosen for its adaptive learning rate and resilience to noisy gradients.
- **Learning Rate Scheduler**: ReduceLROnPlateau, which decreases the learning rate when validation loss plateaus.
- **Batch Size**: 32, balancing GPU memory constraints and stable gradient estimation.
- **Epochs**: 50, sufficient to ensure convergence on synthetic data.
- **Validation Split**: 20% of the dataset is held out to monitor generalization performance.

3 Results

3.1 Training Curves

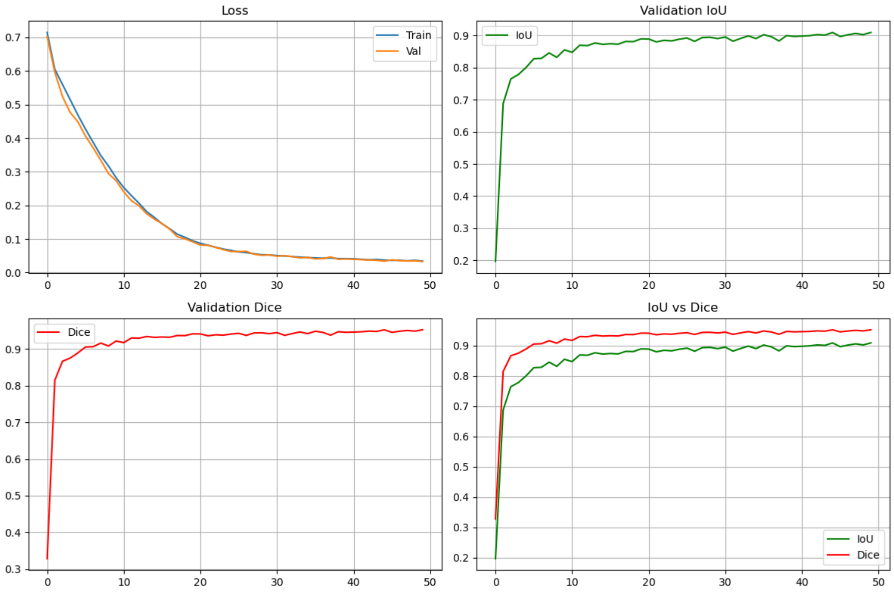

Figure 1: Training and validation metrics across 50 epochs. Loss steadily decreases, and IoU/Dice scores improve consistently, indicating successful learning and robust region-level reconstruction.

3.2 Qualitative Evaluation

Representative samples are shown below to illustrate the network's performance in reconstructing masks from noisy inputs:

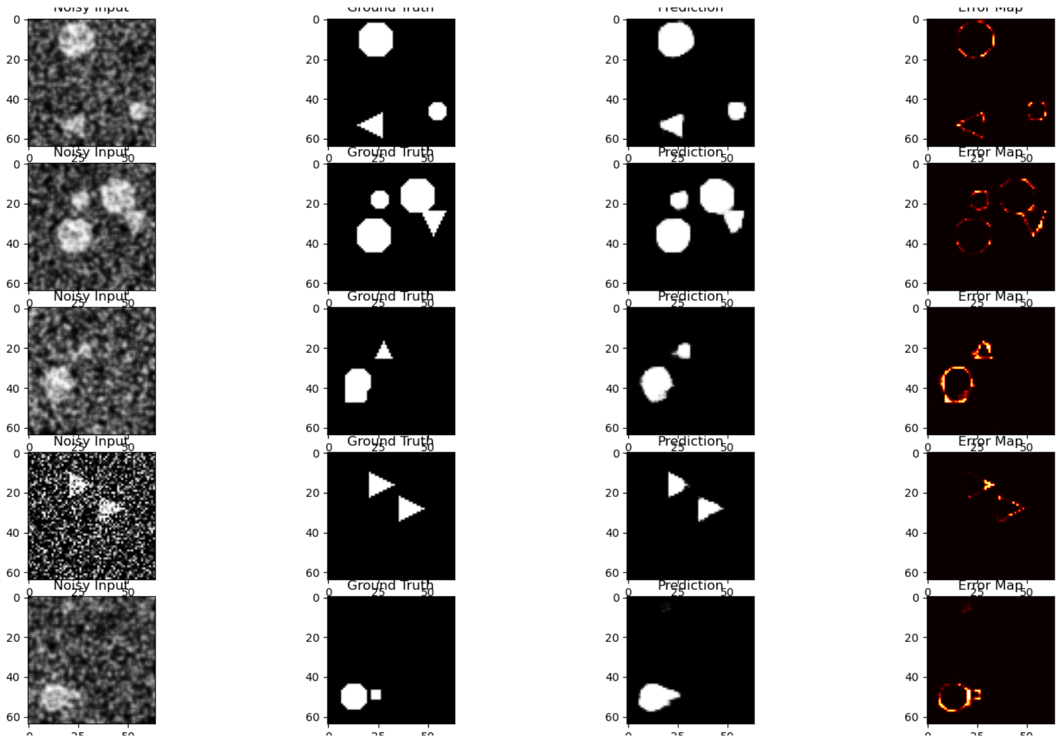

Figure 2: Examples include noisy inputs, ground truth masks, network predictions, and error maps, highlighting the high fidelity of the reconstructions.

3.3 Quantitative Metrics

- Final Validation IoU: 0.909
- Final Validation Dice: 0.952
- Mean BCE: 0.033

These results demonstrate that the model accurately reconstructs clean masks even under stochastic noise.

4 Future Work and Extensions

Several directions could enhance the current approach, particularly if real data were available:

4.1 Advanced Denoising Techniques

- **Diffusion-Based Denoising**: Learn the reverse process of noise addition, potentially improving robustness to complex, real-world noise patterns.

- **Adversarial Training**: Integrate a discriminator in a GAN framework to refine mask boundaries and improve perceptual quality.

4.2 Architectural Enhancements

- **Attention Mechanisms**: Incorporate self-attention or squeeze-and-excitation blocks to capture long-range dependencies in complex masks.
- **Multi-Scale Features**: Use feature pyramid networks or multi-scale inputs to handle objects of varying sizes.

4.3 Training Strategy Improvements

- **Progressive Noise Training**: Start with simple synthetic noise and gradually introduce more realistic variations.
- **Multi-Task Learning**: Simultaneously train on related tasks such as boundary detection to improve feature representations.

4.4 Real-World Considerations

- **Domain Adaptation**: Apply techniques to reduce the gap between synthetic and real data, such as style transfer or adversarial domain alignment.
- **Augmentation for Real Data**: Include realistic transformations like lighting changes, occlusions, or sensor-specific noise.

4.5 Deployment Considerations

- **Uncertainty Estimation**: Quantify model confidence for practical decision-making.
- **Efficiency**: Explore pruning or quantization for deployment on limited-resource devices without degrading performance.

Overall, this methodology provides a detailed, generalizable approach to mask denoising, which can be extended to real-world segmentation tasks with additional refinements.