Adaptive Noise-Type Routing for CNN Denoising

Team: Zachary De Aguiar, Alexander Green , William Moulton, and Zachary Wood

Motivation and Problem Statement

Real-world images are often degraded by diverse noise types such as Gaussian, salt-and-pepper, Poisson, or compression artifacts. Traditional CNN-based denoisers are typically trained for a single noise model, which limits their robustness and generalization. A unified model struggles to adapt to different or mixed noise conditions. This project addresses the problem by developing a noise-type adaptive CNN that identifies the noise class and routes the input through specialized denoising branches.

Research Objectives

1. Develop a noise-type adaptive CNN framework that predicts the noise type present in an input image and applies a specialized denoising branch tuned for that type.
2. Evaluate the denoised images by comparing directly to the corresponding clean ground-truth images (original, noise-free dataset images)
3. Test the model's performance across multiple noise types (Gaussian, salt-and-pepper, Poisson, and potential motion blur or JPEG artifacts) and mixed-noise settings.
4. Analyze whether adaptive routing improves reconstruction quality compared to a single unified model when evaluated against clean ground-truth images.

State of the Art, and your Key Novelty

1. **Noise-type awareness:** Directly measures the denoiser's ability to reconstruct clean images, eliminating reliance on comparisons with other models.

2. **Adaptive multi-branch design:** Dynamically routes inputs to noise-type-specific modules or blends outputs, optimized purely for reconstruction quality. This places a focus on generalization without depending on a single fixed-architecture denoiser.

Impact, Outcomes, Achievements

1. **Impact:** Shows that noise-type adaptive CNNs can more effectively recover clean images under varied noise conditions and provides a transparent, ground-truth–based evaluation framework for denoising research.

2. **Outcomes:** A trained adaptive denoising system evaluated on image datasets with multiple synthetic noise types; results include metrics and visual comparisons to original clean images.

3. **Achievements:** End-to-end implementation of a noise-type classifier with adaptive denoising branches, creation of a reproducible benchmark with quantitative and qualitative results.

Action Plan

Phase 1 (Sep 12 – Oct 1):

Goal: Setup and prototype

- Set up environment and dataset loaders (MNIST, CIFAR-10/100, STL-10, etc)
- Implement synthetic noise generators (Gaussian, salt-and-pepper, Poisson, etc)
- Train baseline denoisers and begin implementation of noise-type classifier with basic routing

Deliverable: Prototype pipeline with sample denoised outputs

Phase 2 (Oct 2 – Nov 6):

Goal: Build full adaptive model and run experiments

- Implement multi-branch adaptive CNN with routing/blending mechanism
- Train model across multiple noise types and datasets
- Evaluate denoised images against clean ground-truth

Deliverable: Trained adaptive model and experimental results

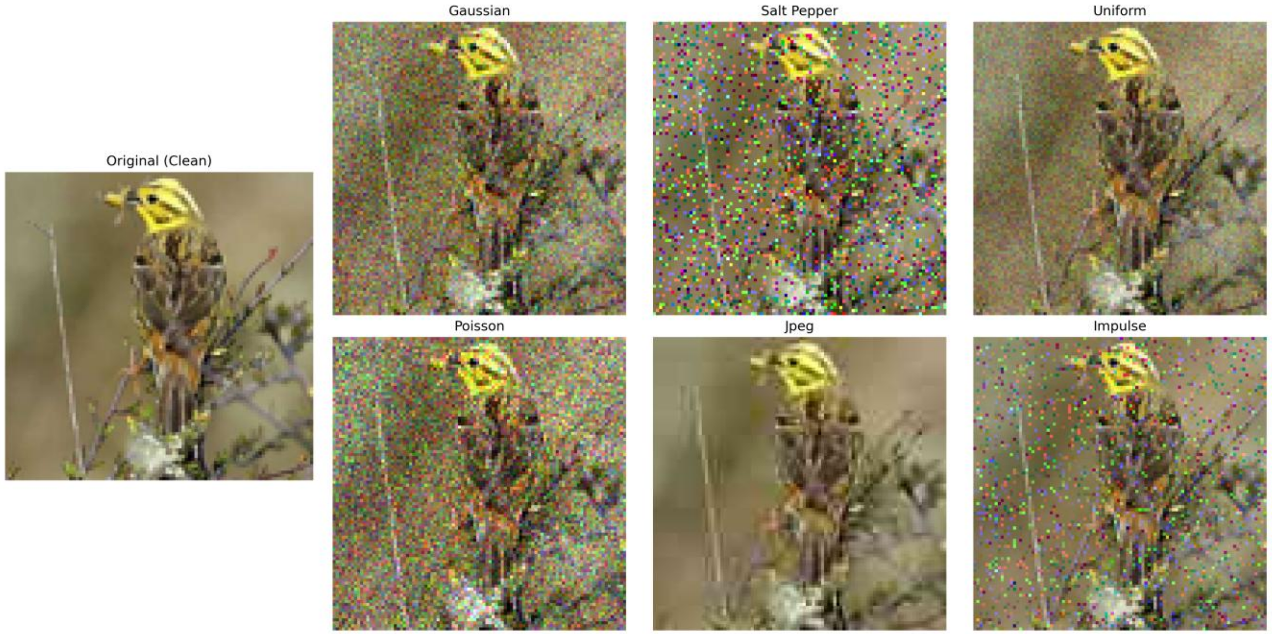
Phase 3 (Nov 7 – Dec 2):

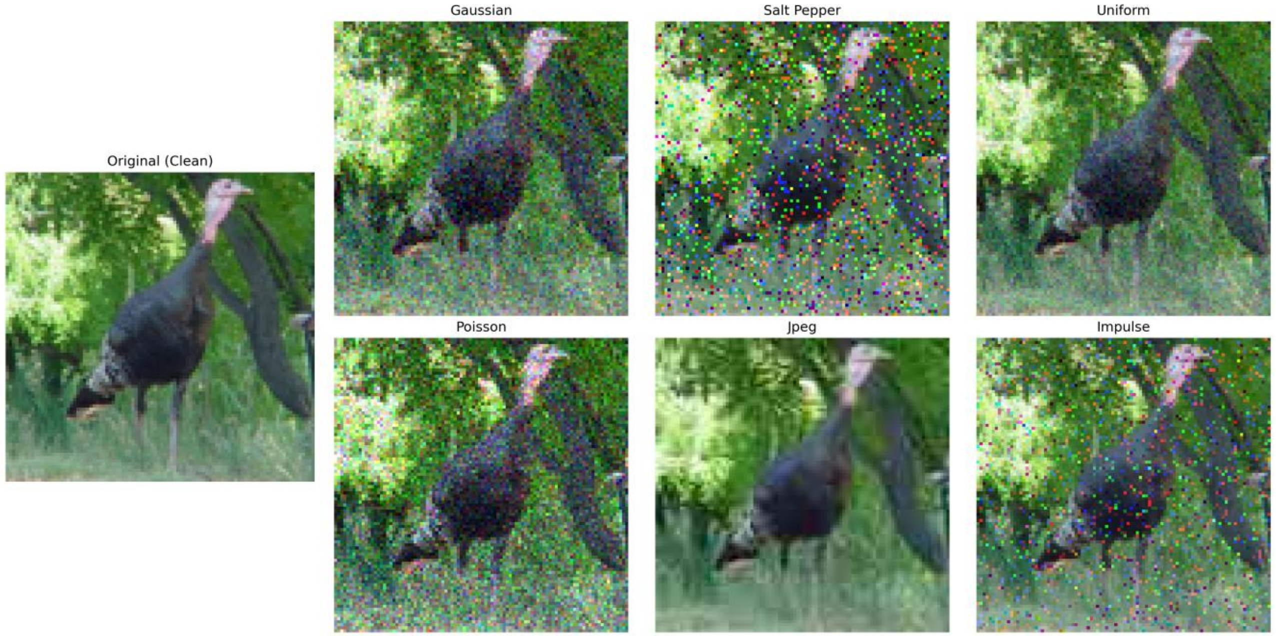
Goal: Analyze results, write report, and prepare presentation

- Analyze quantitative & qualitative results; generate figures (confusion matrix, error curves, etc)
- Write final technical report/paper with methods, experiments, results, and discussion
- Finalize code repository and pre-trained models (README.md, etc)
- Prep presentation

Deliverable: Complete report, reproducible code, and presentation ready

Results so far and path forward





Our first 2 goals of phase 1 were to set up dataset loaders for MNIST, CIFAR-10/100, STL-10, and then to implement synthetic noise generators. We achieved both of these goals.

We currently have a dedicated file which implements dataset loaders for all 4 of the aforementioned datasets.

As for the noise generator, we have included 2 examples of the different types of noise. We implemented the following noise types:

- Gaussian
- Salt and Pepper
- Uniform
- Poisson
- JPEG Compression
- Impulse

Results so far and path forward

Our final goal for phase 1 was to train a baseline denoiser and begin the implementation of noise-type classifier for basic routing.

The images to the right show the performance of our baseline denoiser. This shows the original clean image, the image after we have applied noise, and then the final denoised image. As you can see, it is fairly effective already.

Currently we are working on our implementation of the noise-type classifier for routing. Once this is complete, we will be able to use the output to route or software the correlated noise-specific model