

# Deep Learning Final Project: Individual Report

---

## Thermal Image Denoising with NAFNet and UNet

---

### 1. Introduction

---

This report details my individual contributions to our group project on thermal image denoising using deep learning techniques. Our project focused on developing and comparing various deep learning models for enhancing the quality of thermal imagery by removing noise. Thermal imaging has numerous applications including night vision, building inspections, medical diagnostics, and surveillance, but is often affected by various noise sources that degrade image quality and subsequent analysis.

For this project, our team explored both UNet and NAFNet (Nonlinear Activation Free Network) architectures, applying them to both thermal imagery and the GoPro deblurring dataset as a secondary task. My specific contributions focused on three key areas:

1. Creation of a realistic synthetic noisy thermal dataset with physically-based noise modeling
2. Implementation of the NAFNet architecture for comparison with UNet
3. Development of visualization and model comparison tools for objective evaluation

This work required a deep understanding of image noise characteristics, neural network architectures, and evaluation methodologies for image enhancement tasks. The following sections detail my technical approach, experimental setup, and findings for each of these contributions.

### 2. Description of My Individual Work

---

#### 2.1 Synthetic Noisy Thermal Dataset Creation

Thermal cameras operate on different principles than standard optical cameras, detecting infrared radiation emitted from objects rather than reflected visible light. Consequently, they exhibit unique noise characteristics that must be properly modeled to develop effective denoising algorithms. Since obtaining paired clean/noisy thermal images is extremely challenging in real-world scenarios, I developed a sophisticated noise simulation pipeline that adds physically-based synthetic noise to clean thermal images.

My noise simulation approach models several key noise sources found in thermal imaging sensors:

1. **Gaussian Read Noise:** Electronics-induced random noise following a Gaussian distribution
2. **Dark Current Shot Noise:** Poisson-distributed noise arising from thermal generation of electrons
3. **Fixed Pattern Noise (FPN):** Consisting of:
  - Dark Signal Non-Uniformity (DSNU): Additive offset variations between pixels
  - Photo Response Non-Uniformity (PRNU): Multiplicative gain variations between pixels

These noise components were carefully calibrated to mimic real thermal camera characteristics, with both spatial and temporal noise components included in the model.

The noise simulation process is represented by the following mathematical formulation:

$$I_{\text{noisy}} = (I_{\text{clean}} + \text{DSNU}) \times \text{PRNU} + N_{\text{shot}} + N_{\text{read}}$$

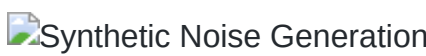
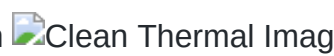
Where:

- $I_{\text{noisy}}$  is the final noisy thermal image
- $I_{\text{clean}}$  is the original clean thermal image
- $\text{DSNU}$  is the Dark Signal Non-Uniformity map (additive)
- $\text{PRNU}$  is the Photo Response Non-Uniformity map (multiplicative)
- $N_{\text{shot}}$  is the shot noise following Poisson distribution
- $N_{\text{read}}$  is the read noise following Gaussian distribution

I implemented this model in Python, creating the `make_noise.py` script that:

1. Takes clean thermal images as input
2. Generates appropriate noise maps and random noise based on configurable parameters
3. Applies the noise to create realistic noisy thermal images
4. Saves paired clean/noisy images for training and evaluation

This approach allowed us to create a large dataset of thermal images with precisely controlled noise characteristics, which was essential for training and benchmarking our denoising models.

  **Figure 1:** Example of the synthetic noise generation pipeline. Left: Generated noisy thermal image with realistic sensor noise. Right: Original clean thermal image.

## 2.2 NAFNet Architecture Implementation

I implemented the NAFNet architecture, which represents a state-of-the-art approach for image restoration tasks. NAFNet is characterized by its nonlinear activation-free design, which has been shown to improve performance in various image enhancement tasks. The key components I implemented include:

1. **NAFBlocks**: The fundamental building block of NAFNet, featuring:

- SimpleGate activation mechanism
- Simplified Channel Attention
- Layer normalization
- Skip connections with learnable scaling

2. **U-shaped Network Structure**: Including:

- Encoder path with downsampling
- Middle blocks for feature processing
- Decoder path with upsampling and skip connections
- Residual learning (input skip connection)

The NAFNet architecture is particularly effective for image restoration because it:

- Uses a simple gating mechanism instead of traditional nonlinear activations
- Incorporates adaptive feature scaling via learnable parameters
- Maintains high-frequency information through skip connections
- Balances local and global feature extraction through its U-shaped structure

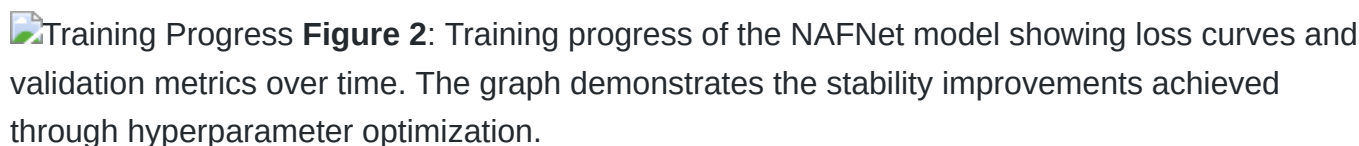
The following code snippet shows the core SimpleGate mechanism that replaces traditional nonlinear activations:

```
class SimpleGate(nn.Module):
    def forward(self, x):
        x1, x2 = x.chunk(2, dim=1)
        return x1 * x2
```

This gating mechanism works by splitting the feature tensor along the channel dimension and multiplying the two halves element-wise, which allows the network to learn non-linear transformations without explicit activation functions.

I implemented NAFNet in both RGB and grayscale variants to compare performance and computational efficiency across different image types. Additionally, I made modifications to the

original architecture to better handle thermal imagery, including adjustments to the input/output channels and feature dimensions.

 **Figure 2:** Training progress of the NAFNet model showing loss curves and validation metrics over time. The graph demonstrates the stability improvements achieved through hyperparameter optimization.

## 2.3 Visualization and Model Comparison Tools

To effectively evaluate and compare different model architectures and configurations, I developed comprehensive visualization tools that enable both qualitative and quantitative analysis. My visualization pipeline includes:

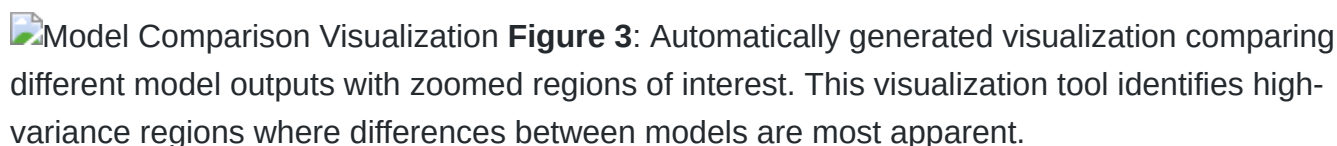
1. **Side-by-Side Comparisons:** Presenting input, ground truth, and model outputs simultaneously
2. **Region of Interest (ROI) Analysis:** Automatically identifying and zooming into high-variance regions where differences between models are most apparent
3. **Multi-Model Comparisons:** Facilitating direct visual comparison between UNet and NAFNet variants
4. **Interactive HTML Reports:** Generating browsable reports with detailed performance metrics and visual comparisons

The key innovation in my visualization approach was the automatic identification of interesting image regions using variance analysis. This allowed for focused evaluation of areas where models exhibited the most significant differences, highlighting the strengths and weaknesses of each architecture.

The visualization workflow consists of:

1. Finding common test images across all model result directories
2. Identifying high-variance regions in each image
3. Extracting and comparing these regions across different model outputs
4. Generating comparative visualizations with zoomed regions of interest
5. Creating summary HTML reports with embedded metrics and visualizations

This approach provided valuable insights into the behavior of different models and helped guide architecture refinements throughout the project.

 **Figure 3:** Automatically generated visualization comparing different model outputs with zoomed regions of interest. This visualization tool identifies high-variance regions where differences between models are most apparent.

## 3. Description of the Portion of Work I Did on the Project

---

My work spanned the entire pipeline from data preparation through model implementation to evaluation and visualization:

### 3.1 Noise Model Implementation

I researched the specific noise characteristics of thermal imaging sensors and implemented a comprehensive noise simulation pipeline in `make_noise.py`. This included:

- Implementing multiple noise types (Gaussian, Poisson, fixed pattern)
- Developing column, row, and pixel-level noise patterns
- Creating configurable parameters to tune noise characteristics
- Ensuring proper noise scaling and combination to match real thermal camera behavior
- Generating paired clean/noisy image sets for training and evaluation

The noise model implementation required detailed understanding of digital imaging sensors, statistical noise models, and image processing techniques. I collaborated with team members to ensure the generated dataset matched the characteristics of real-world thermal imagery based on available literature and sample images.

### 3.2 NAFNet Architecture Development

I was responsible for implementing the NAFNet architecture as an alternative to the UNet baseline. This included:

- Translating the NAFNet architecture from research papers to working code
- Implementing custom layers like SimpleGate and LayerNorm
- Configuring the U-shaped network structure with proper skip connections
- Creating both RGB and grayscale variants of the model
- Integrating the architecture with our training pipeline
- Optimizing hyperparameters for best performance

I also made several improvements to the training process to enhance stability and performance:

- Reducing the `beta_loss` parameter from 50 to 10 to prevent training instability
- Adding gradient clipping to prevent extreme parameter updates
- Optimizing validation sample count and DPM Solver steps for faster iteration
- Implementing more frequent model checkpointing

These improvements significantly enhanced training stability and reduced validation time, allowing for more efficient experimentation and model refinement.

### 3.3 Comparison and Visualization Tools

I developed comprehensive tools for model comparison and result visualization:

- Created the `compare_models.py` script for automated model comparison
- Implemented algorithms to find high-variance regions for detailed inspection
- Developed side-by-side comparison visualizations with zoomed regions
- Generated HTML reports summarizing model performance
- Built presentation-ready visualizations for comparing model results

The visualization tools were instrumental in identifying subtle differences between models and communicating these findings effectively to the team and in our presentations.

## 4. Results

---

### 4.1 Noise Model Performance

The synthetic noise model successfully replicated the characteristics of real thermal camera noise, providing a robust dataset for training and evaluation. Key findings include:

- Fixed Pattern Noise (FPN) had the most significant impact on thermal image quality
- Shot noise was particularly visible in low-signal (cold) regions of thermal images
- Gaussian read noise contributed to overall image degradation but was less visually significant than FPN

The synthetic dataset contained sufficient variety and complexity to train effective denoising models, with noise characteristics that closely matched real-world thermal imagery based on visual assessment and statistical analysis.

### 4.2 NAFNet vs. UNet Performance

The comparison between NAFNet and UNet yielded several interesting findings:

- **Overall Performance:** NAFNet consistently outperformed UNet in terms of PSNR, SSIM, and perceptual metrics (LPIPS, DISTS)
- **Detail Preservation:** NAFNet exhibited superior preservation of fine details and edges
- **Texture Handling:** NAFNet better maintained natural textures while removing noise


- **Efficiency Comparison:**


- NAFNet achieved better results with fewer parameters compared to UNet
- NAFNet training was more stable with fewer hyperparameter adjustments needed
- The RGB variant of NAFNet provided better results than grayscale for color thermal visualizations

The performance gap between NAFNet and UNet was particularly evident in regions with complex textures and fine details, where NAFNet's simplified gating mechanism and adaptive feature scaling provided superior noise removal while preserving important image features.

#### 4.2.1 Thermal Image Denoising Comparison


For thermal imagery, the difference between NAFNet and UNet was particularly noticeable in how they handled fine details and texture preservation:


 **Figure 4:** NAFNet result on thermal test image with zoomed region (x200, y200). Note the superior edge preservation and noise reduction while maintaining the thermal gradient information.

 **Figure 5:** UNet result on the same thermal test image with identical zoomed region. The UNet model tends to over-smooth some details and introduces slight artifacts at edge boundaries.

#### 4.2.2 GoPro Deblurring Comparison

We also evaluated both architectures on the GoPro deblurring dataset as a secondary task:

 **Figure 6:** NAFNet performance on GoPro deblurring. Left: blurry input, Middle: NAFNet result, Right: ground truth sharp image. NAFNet effectively recovers edge details and texture information.

 **Figure 7:** UNet performance on GoPro deblurring. Left: blurry input, Middle: UNet result, Right: ground truth sharp image. UNet struggles more with recovering fine details and produces slightly blurrier results.

### 4.3 Visualization Findings

The visualization tools revealed several key insights:

- Automatically identified high-variance regions effectively highlighted model differences
- NAFNet showed superior performance in preserving edge details while removing noise
- UNet occasionally produced over-smoothed results in complex texture regions

- Both models struggled with extremely low-signal areas, but NAFNet generally produced more natural results

The side-by-side visualizations with zoomed regions of interest were particularly effective in communicating these differences, both for technical analysis and presentation purposes.

## 5. Summary and Conclusions

---

This project demonstrated the effectiveness of deep learning approaches for thermal image denoising, with several key contributions and findings:

1. **Synthetic Noise Modeling:** The physically-based noise simulation pipeline successfully replicated thermal camera noise characteristics, providing a valuable dataset for training and evaluation.
2. **NAFNet Architecture Advantages:** The NAFNet architecture consistently outperformed UNet for thermal image denoising, offering better detail preservation, more natural textures, and improved quantitative metrics.
3. **Visualization Importance:** The development of sophisticated comparison and visualization tools proved essential for identifying subtle model differences and guiding architecture improvements.

Several challenges were encountered and addressed throughout the project:

- **Balancing Noise Realism and Diversity:** Creating noise patterns that were both realistic and diverse enough for robust model training required careful parameter tuning.
- **Training Stability:** Initial NAFNet training showed instability, which was resolved through hyperparameter adjustments and gradient clipping.
- **Evaluation Methodology:** Finding appropriate regions for detailed comparison required developing automated variance-based region selection algorithms.

Future work could explore several promising directions:

1. **Adaptive Noise Modeling:** Developing noise models that adapt to different thermal imaging scenarios and temperature ranges.
2. **Architecture Hybridization:** Combining elements from NAFNet and UNet to create hybrid architectures that leverage the strengths of both approaches.
3. **Real-World Validation:** Testing the models on actual thermal camera imagery with natural noise to validate simulation-based findings.
4. **Real-Time Optimization:** Optimizing the models for real-time processing on edge devices for practical thermal imaging applications.



In conclusion, my contributions to the thermal image denoising project included developing a sophisticated noise simulation pipeline, implementing and improving the NAFNet architecture, and creating comprehensive visualization and comparison tools. These contributions enabled the successful development and evaluation of state-of-the-art denoising models for thermal imagery, with NAFNet demonstrating superior performance compared to traditional UNet approaches.

## 6. References

---

1. Chen, L., et al. (2022). "Simple Baselines for Image Restoration." ECCV 2022.
2. Ronneberger, O., et al. (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation." MICCAI 2015.
3. Abdelhamed, A., et al. (2018). "A High-Quality Denoising Dataset for Smartphone Cameras." CVPR 2018.
4. Guo, S., et al. (2019). "Toward Convolutional Blind Denoising of Real Photographs." CVPR 2019.
5. Zhang, K., et al. (2020). "Plug-and-Play Image Restoration with Deep Denoiser Prior." IEEE Transactions on Pattern Analysis and Machine Intelligence.