BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Project report on

Image-Restoration using Multistage U-Net Architecture

Submitted

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Mohammed Sadiq Z Pattankudi	01FE22BCI027
Samarth Uppin	01FE22BCI008
Abdul Rafay Attar	01FE22BCI001
Kashish Jewargi	01FE22BCS031

Under the guidance of
Dr. Meena S M
Professor and Dean of Faculty

School of Computer Science and Engineering

KLE Technological University, Hubballi

2025-2026

KLE Technological University
Creating Value
Leveraging Knowledge
BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2025-26

CERTIFICATE

This is to certify that project entitled "Image-Restoration using Multistage U-Net Architecture" is a bonafide work carried out by the student team Mohammed Sadiq Z Pattankudi (01FE22BCI027), Samarth Uppin (01FE22BCI008), Abdul Rafay Attar (01FE22BCI001), and Kashish Jewargi (01FE22BCS031), in partial fulfillment of the completion of 7th semester B.E. course during the year 2024–2025. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Head, CSE(AI)

Dr. Meena S M

Dr. Narayan D. G.

External Viva-Voce

Name of the examiners Signature with date

1 ______________________ ______________

2 ______________________ ______________

ABSTRACT

Our work presents a multi-stage U-Net-based architecture for high-quality image restoration, specifically designed for removing noise from RGB images. Current image denoising models are often limited because of high computational cost, making them difficult to run on resource-constrained devices, and their performance is further limited by the scarcity of diverse and high-quality denoising datasets. The proposed model consists of two sequential U-Net stages, each integrated with Convolutional Block Attention Modules (CBAM) to enhance feature representation by focusing on informative spatial and channel dimensions, followed by a refinement block. Each stage predicts a residual correction, progressively refining the input image by subtracting the accumulated noise and restoring finer details. The encoder–decoder framework of each U-Net captures hierarchical features, while CBAM selectively emphasizes critical regions for effective noise suppression. The model is trained on the custom Smartphone Image Denoising Dataset (SIDD) using a hybrid loss function that combines Lp loss with a weighted SSIM loss to ensure both pixel-wise accuracy and perceptual quality. Experimental results show that our proposed method achieves a Peak Signal-to-Noise Ratio (PSNR) of 33.63 dB and a Structural Similarity Index (SSIM) of 0.87 on the SIDD dataset. Our future work is focused on deploying the proposed model on edge devices, such as smartphones and surveillance cameras.

Keywords: Image Restoration, Multi-stage U-Net, CBAM, Residual Learning, Image Denoising, PSNR, SSIM, Attention Mechanism, Deep Learning.

ACKNOWLEDGEMENT

We would like to thank our faculty and management for their professional guidance towards the completion of the Senior Design Project work. We take this opportunity to thank Dr. Ashok Shettar, Pro-Chancellor, Dr. P.G. Tewari, Vice-Chancellor and Dr. B.S. Anami, Registrar for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Dean of Faculty, SoCSE, Dr. Narayan D. G, Professor and Head, Department of CSE(AI) for having provided us direction and facilitated for enhancement of skills and academic growth.

We sincerely thank our guide Dr. Meena S. M, Professor and Dean of Faculty, Department of Computer Science and Engineering for her guidance, inspiration and wholehearted co-operation during the course of completion.

We sincerely thank our project co-ordinators Dr. P.G Sunita Hiremath for their support, inspiration and wholehearted co-operation during the course of completion.

Our gratitude will not be complete without thanking the Almighty God, our beloved parents, our seniors and our friends who have been a constant source of blessings and aspirations.

Mohammed Sadiq Z Pattankudi 01FE22BCI027
Samarth Uppin 01FE22BCI008
Abdul Rafay Attar 01FE22BCI001
Kashish Jewargi 01FE22BCS031

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Image restoration is a key challenge in computer vision and image processing, focused on recreating a high-quality image from its degraded version. Degradation can occur due to sensor noise, motion blur, or environmental factors during image capture, storage, or transmission [1, 2]. As imaging devices are now widely used in fields like consumer electronics, autonomous vehicles, medical imaging, and surveillance, there is a growing need for fast and reliable restoration methods [3, 4]. Producing high-quality restored images is crucial, as it directly impacts the accuracy of automated systems and human decision-making in critical applications, from medical diagnosis to autonomous navigation [5, 6].

Traditional image restoration techniques, such as Non-Local Means (NLM) [7] and Block-Matching 3D filtering (BM3D) [8], leverage hand-crafted priors and self-similarity across image patches to reduce noise. While effective under controlled conditions, these methods tend to underperform in the presence of complex, signal-dependent noise patterns, especially in real-world scenarios. The advent of deep learning has significantly advanced the computer vision field. Supervised models such as DnCNN [1], IRCNN [9], and FFD-Net [10] have demonstrated notable improvements by learning end-to-end mappings from noisy to clean images. However, most of these models assume synthetic Gaussian noise during training, limiting their generalizability to real-world degradations. Additionally, they often lack mechanisms to adaptively emphasize informative regions, which is critical for preserving structural details and texture fidelity.

The development of realistic benchmarks such as the Smartphone Image Denoising Dataset (SIDD) [11] has marked a turning point in real-world image restoration research. By providing real noisy-clean image pairs captured from smartphone cameras, SIDD allows models to learn complex noise patterns and evaluate performance under practical conditions [12]. Due to its high relevance and quality, SIDD has become one of the most widely used datasets in image denoising research. Despite these advancements, challenges such as noise variability, texture preservation, and computational efficiency remain open problems.

To overcome the challenges posed by limited datasets and low computational resources, we propose a multi-stage U-Net architecture enhanced with Convolutional Block Attention Modules (CBAM) for robust real-world image restoration. The framework consists of two cascaded U-Net stages followed by a refinement block, where each stage learns a residual correction that progressively improves image quality. The integration of CBAM facilitates both spatial and channel-wise attention, enabling the network to focus on informative regions while suppressing noise. This design ensures the preservation of global structural information as well as fine-grained details throughout the restoration process.

1.1 Sustainable Development Goal (SDG)

Our project aligns with the Sustainable Development Goal 9 which focuses on building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation. It is particularly relevant to Target 9.5 that emphasizes enhancing scientific research and strengthening the technological capabilities of industrial sectors. By developing an efficient multi-stage U-Net architecture capable of performing high-quality image restoration on resource-constrained devices, our project contributes to advancements in computational imaging and supports innovation in low-power AI systems. Our work also aligns with Indicator 9.5.2 by contributing to research output and promoting growth in technology-driven research and development activities.

1.2 Motivation

Recent advancements in deep learning and computer vision have increased our dependency on high-quality image data for AI applications, including medical imaging, autonomous systems, and surveillance, which highlights the need for advanced image restoration methods, as noise and degradation can severely hinder feature extraction and degrade overall model performance.

Deploying AI models on embedded and edge devices with limited computational resources requires innovative image denoising methods that balance accuracy and efficiency. Our project addresses the above challenge by applying global unstructured pruning to our multi-stage U-Net architecture enhanced with Convolutional Block Attention Modules (CBAM), facilitating selective feature enhancement and efficient noise suppression. Our goal is to elevate image quality in resource-constrained environments, ensuring that high-quality visual data remains impactful across various applications.

1.3 Literature Survey

Image denoising has a long history rooted in traditional signal processing techniques that leverage prior knowledge and image statistics. Non-Local Means (NLM) [7] and Block-Matching 3D filtering (BM3D) [8] are two landmark methods that rely on patch-based redundancy and self-similarity. NLM averages similar patches across the image to suppress noise, while BM3D enhances the above process using collaborative filtering in a 3D transform domain. Though effective in controlled settings, these methods often degrade under complex, real-world noise conditions where handcrafted priors fail to adapt to varying noise characteristics or preserve subtle image details.

The rise of deep learning led to a paradigm shift in denoising strategies. DnCNN [1] introduced a convolutional neural network trained end-to-end on Gaussian noise, offering substantial improvements in PSNR and generalization across different noise levels. IRCNN

[9] combined residual learning with image priors for iterative denoising, while FFDNet [10] incorporated noise-level maps to enable dynamic control over denoising strength. Despite their performance, these models generally assume noise is spatially invariant and Gaussian, limiting their effectiveness in real-world conditions where noise is often signal-dependent and structured.

To overcome these limitations, attention mechanisms were introduced to improve feature representation and enhance structural preservation. Self-guided approaches such as ADNet [13] and SADNet [14] allow the network to adaptively focus on salient image regions, leading to better edge sharpness and texture recovery. These techniques incorporate spatial or channel-wise attention to identify important features, mitigating the tendency of earlier CNN-based models to oversmooth fine details. However, attention-heavy architectures are often computationally intensive and may suffer from high latency or overfitting.

The availability of realistic datasets such as the Smartphone Image Denoising Dataset (SIDD) [11] marked a significant milestone by providing real noisy-clean image pairs from consumer-grade sensors. It enables training and evaluation under practical noise conditions, uncovering the limitations of models trained purely on synthetic data. Studies have shown that deep models, although accurate in clean lab settings, often struggle with generalization on real-world images due to unpredictable noise patterns and the need for preserving texture and contrast [15, 16].

More recent work has shifted toward richer architectures that combine local and global context. For example, Restormer [17] leverages transformer blocks to capture long-range dependencies for improved denoising. Meanwhile, plug-and-play networks [18] and pre-trained diffusion models [19] attempt to generalize denoising by incorporating external priors or generative guidance. These approaches have shown promising results but often come with high computational costs and complex training regimes, making them less suitable for lightweight or real-time applications.

In light of these developments, there is a clear need for a denoising model that combines the strengths of convolutional architectures—such as spatial efficiency and hierarchical feature extraction—with the adaptability of attention mechanisms. Our proposed approach addresses the above gap by integrating Convolutional Block Attention Modules (CBAM) within a multi-stage U-Net framework. The multi-stage design allows for progressive refinement through residual learning at each stage, enhancing detail restoration in an iterative manner. CBAM modules guide the network's attention across spatial and channel dimensions, helping it focus on informative features while suppressing irrelevant noise. The above synergy results in a model that is both robust to real-world noise and efficient in computation, achieving high fidelity restoration on the SIDD dataset.

1.4 Problem Statement

In real-world imaging conditions, captured images often contain complex and unpredictable noise arising from sensor limitations, lighting variations, motion, and environmental factors. Traditional denoising techniques and many deep learning models do not generalize well to these conditions because they are typically trained on small and non-diverse datasets and are designed for simplified synthetic noise. Existing deep learning–based restoration models also tend to incur high computational costs, making them unsuitable for deployment on resource-limited devices. As a result, these methods often fail to preserve the fine textures and structural details required for producing high-quality restored images. To address these limitations, we complement existing benchmarks by creating our own custom dataset that captures more diverse and realistic noise patterns. Our goal is to develop a robust and resource-efficient image restoration model capable of effectively handling real-world noise, preserving both global structure and fine-grained details, and generating high-quality restored images suitable for practical applications.

1.5 Problem Analysis

To effectively address the challenges identified in the problem statement, our project adopts two key software engineering design principles: **Modular Design** and **Separation of Concerns (SoC)**. These principles are essential for managing the complexity of a multi-stage image restoration system. By applying Modular Design, we structure the model into clearly defined components such as the U-Net stages, refinement block, CBAM modules, and the custom dataset pipeline. This allows each part to be developed, tested, and improved independently, supporting flexibility and easy integration of enhancements. Separation of Concerns further strengthens the architecture by ensuring that each component focuses on a specific responsibility—for example, feature extraction, attention refinement, residual correction, or data preparation—resulting in a cleaner, more organized system. Together, these principles improve maintainability, scalability, and experimentation efficiency, enabling the model to better meet the requirements of real-world image restoration.

1.5.1 Design Principle 1 – Modular Design

Modular Design emphasizes building a system as a set of distinct, independent components that can be developed, tested, and improved separately. This principle strongly aligns with our proposed image restoration framework. The architecture consists of multiple independently functioning modules—U-Net Stage 1, U-Net Stage 2, the refinement block, CBAM modules, and the custom dataset pipeline. Each module performs a well-defined role within the overall process, allowing for flexibility and ease of modification. For example, one can upgrade the attention mechanism, adjust the refinement block, or experiment with alternative U-Net configurations without requiring major changes to the remaining system. This modular approach simplifies debugging, enhances maintainability, and supports future scalability or experimentation.

1.5.2 Design Principle 2 – Separation of Concerns (SoC)

Separation of Concerns ensures that different functionalities within a system are isolated so that each part focuses on a specific responsibility. This principle is essential for our image restoration model because the process involves multiple distinct tasks, including noise modeling, feature extraction, attention-based refinement, residual correction, and performance evaluation. By clearly separating these concerns, the system becomes easier to understand, maintain, and enhance. For instance, the CBAM module is solely responsible for improving feature attention, while the U-Net backbone manages structural reconstruction, and the custom dataset module handles data diversity. Changes in one component—such as extending the dataset or adjusting the attention mechanism—do not interfere with the operation of others. This results in a cleaner, more organized architecture that supports reliable model development and experimentation.

1.5.3 Scope and Constraints

Scope

- The project focuses on developing a multi-stage U-Net architecture enhanced with CBAM for real-world RGB image denoising.
- The system includes the creation of a custom noisy image dataset to improve model generalization and training diversity.
- The model is designed to operate efficiently on systems with limited computational resources while maintaining high restoration quality.
- The scope covers training, evaluation, and comparison of the proposed model using benchmarks such as the SIDD dataset.
- The project emphasizes modular design and separation of concerns to ensure maintainability and easy experimentation.

Constraints

- The integration of CBAM with the U-Net architecture increases model complexity, making hardware-level optimizations such as quantization more difficult for edge environments.
- The performance of the model depends on the availability of GPU resources for training due to the computational cost of attention modules.

- The creation of a custom dataset requires significant time for data collection, preprocessing, and annotation.
- Memory limitations may restrict the batch size and resolution used during training.
- Real-world noise patterns are highly diverse, which may limit the model's ability to generalize beyond the dataset used.

1.6 Objectives

The primary objectives of our research are:

- To design and implement a multi-stage U-Net architecture tailored for real-world RGB image denoising, enabling progressive refinement through sequential residual learning.
- To integrate Convolutional Block Attention Modules (CBAM) within the network to enhance channel-wise and spatial feature discrimination, improving texture preservation and structural consistency.
- To create a custom real-world noisy image dataset that captures diverse and practical noise patterns to support more effective model training and evaluation.
- To improve the computational efficiency of the model, ensuring it can operate effectively on systems with limited computing resources without compromising restoration quality.
- To evaluate the model using realistic benchmarks such as SIDD, targeting a Peak Signal-to-Noise Ratio (PSNR) exceeding 30 dB and competitive SSIM values as indicators of successful restoration performance.

Chapter 2

REQUIREMENT ANALYSIS

The Requirement Analysis(RA) defines the functional and non-functional requirements for implementing a deep learning-based image denoising system using a multi-stage U-Net architecture enhanced with Convolutional Block Attention Modules (CBAM). Our architecture uses progressive residual learning across multiple U-Net stages to iteratively reduce noise, while CBAM modules help the network focus on important spatial and channel-wise features for improved restoration. The RA ensures the system meets critical expectations regarding denoising quality, model efficiency, and real-world applicability. It outlines the system's performance goals, integration needs, and operational constraints to guide effective design, development, and deployment in domains such as mobile photography, medical imaging, and surveillance.

2.1 Functional Requirements

The functional requirements define the essential operations and behaviour that the proposed image restoration model must perform to meet its intended objectives. These requirements outline the core architectural components, training process, and inference capabilities necessary for effective noise reduction. The functional requirements are as follows:

1. **Input Handling:** The system must accept noisy RGB images from benchmark datasets (e.g., SIDD) using a standardized data-loading and preprocessing pipeline.
2. **Architecture Design:** The model must consist of two cascaded U-Net stages followed by a refinement block, where each stage performs residual correction to progressively reduce noise levels.
3. **Attention Integration:** Each U-Net stage must incorporate Convolutional Block Attention Modules (CBAM) to enable channel-wise and spatial attention during feature extraction and reconstruction.
4. **Inference Capability:** The trained model must support high-resolution image denoising during inference and achieve performance consistent with expected PSNR and SSIM benchmarks.

2.2 Non Functional Requirements

The non-functional requirements specify the performance, quality, and reliability standards that our system must satisfy beyond its core functionality. These requirements ensure that the model operates efficiently and maintains consistent accuracy under practical conditions. The non-functional requirements are as follows.

1. **Performance:** The model should achieve inference latency under 1000 milliseconds per image on GPU-enabled environments like Google Colab.

2. **Accuracy:** Maintain a PSNR of at least 30 dB and SSIM of 0.75 or higher on the SIDD dataset for robust denoising performance.

2.3 Hardware Requirements

The hardware setup consists of the Google Colab environment with:

- Access to NVIDIA Tesla T4 or P100 GPUs with 12–16 GB GPU memory.
- Virtual CPUs (2–4 cores) sufficient for preprocessing and model orchestration.
- 12–16 GB of system RAM provided by the Colab runtime.
- Persistent cloud storage via Google Drive integration for datasets, model checkpoints, and outputs.

2.4 Software Requirements

The software environment for our project includes:

- PyTorch (version 1.13.0 or later) for deep learning model development and GPU acceleration.
- Torchvision for image transformations, datasets, and utilities.
- NumPy and OpenCV for image preprocessing and analysis.
- Matplotlib for plotting training metrics and visualization.
- tqdm for progress visualization during data loading and training.
- Weights & Biases (wandb) for optional experiment tracking.
- pytorch-ssim for SSIM loss computation.
- CUDA Toolkit compatible with Colab's GPU environment (typically CUDA 11.x).
- Jupyter Notebook environment provided natively by Google Colab for interactive experimentation.

Chapter 3

SYSTEM DESIGN

This chapter outlines the architectural framework and design principles of our project.

3.1 Architectural Framework

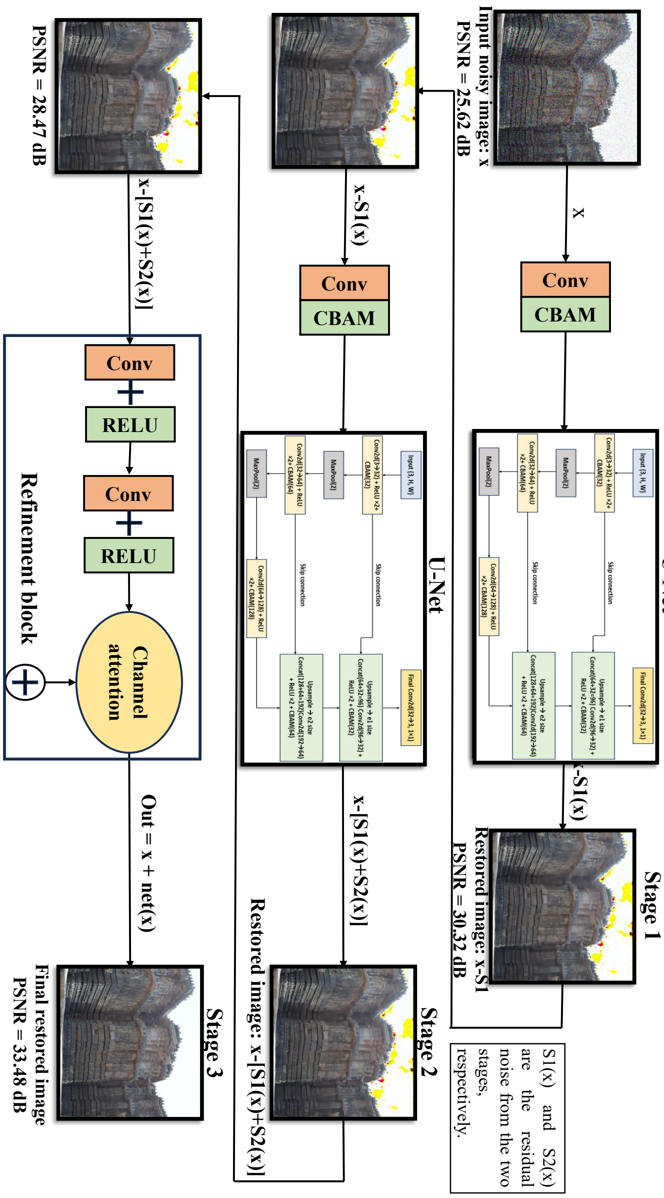

Figure 3.1: Overview of the proposed multistage U-Net architecture for image restoration.

As shown in the figure 3.1, the model consists of two sequential U-Net stages with CBAM modules followed by a refinement block that progressively refines the noisy input to produce a high-quality denoised output. It is designed to restore noisy images progressively, and is trained using a loss function that balances perceptual quality and pixel-level accuracy.

Convolutional Block Attention Module (CBAM)

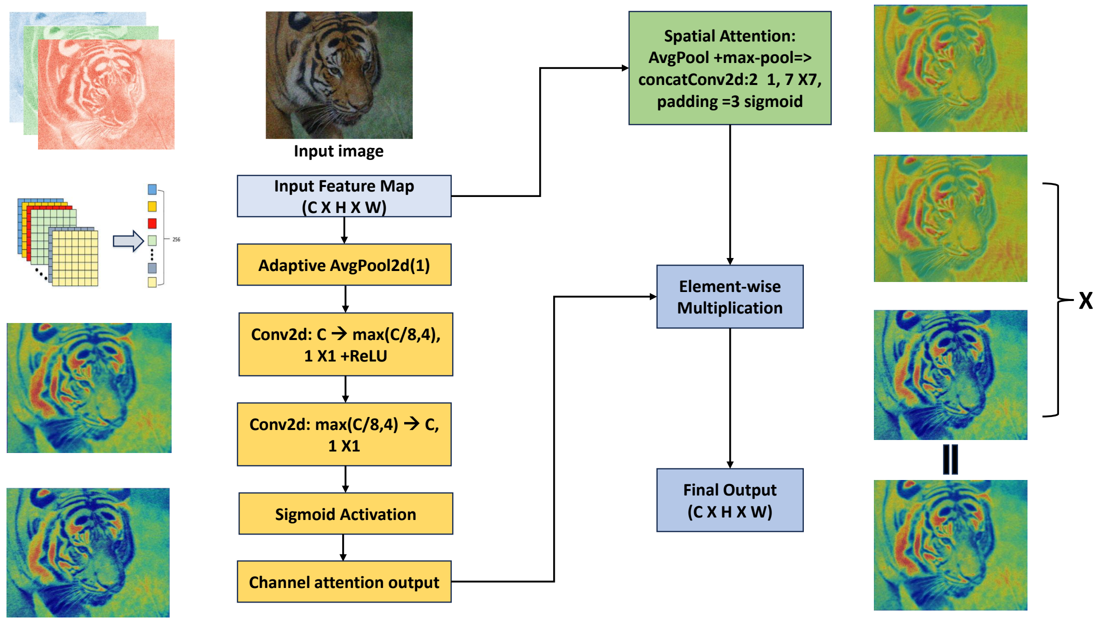

Figure 3.2: CBAM architecture

CBAM [20] enhances feature representations using a combination of channel and spatial attention. The channel attention mechanism is defined in Eq. (3.1), where average and max pooling are followed by a shared Multi-Level Perceptron (MLP) and sigmoid activation:

$$M_c(F) = \sigma\left(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))\right) \tag{3.1}$$

The spatial attention is applied to the channel-refined features, as described in Eq. (3.2):

$$M_s(F') = \sigma\left(f^{7\times 7}\left([\text{AvgPool}(F'); \text{MaxPool}(F')]\right)\right) \tag{3.2}$$

The final attention-weighted output is obtained by element-wise multiplication of the spatial attention map with the input, as shown in Eq. (3.3):

$$F'' = M_s(F') \otimes F' \tag{3.3}$$

U-Net module

The proposed model consists of two U-Nets and one refinement stage arranged sequentially to iteratively refine the input image. Each stage of the U-Net is enhanced with CBAM attention modules embedded within its convolutional blocks to adaptively emphasize informative features.

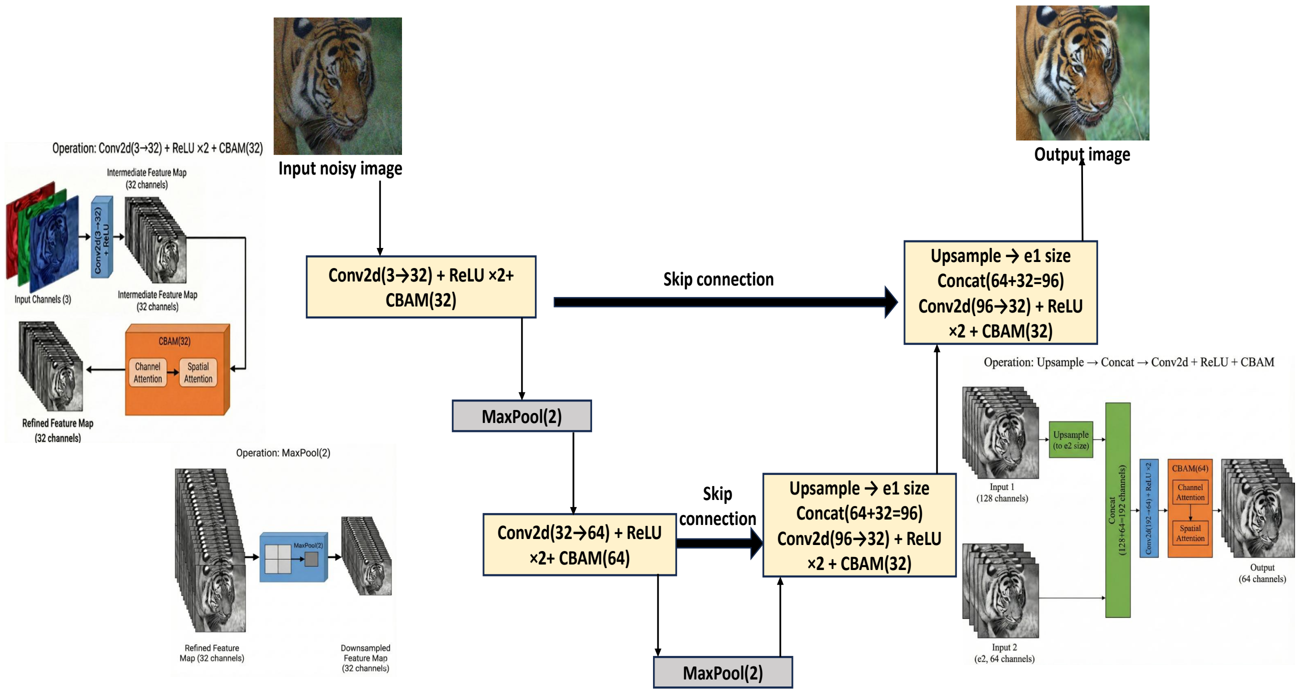

Figure 3.3: U-Net module architecture of our model.

Encoder Path

In each stage U-Net stage, the encoder consists of three convolutional blocks. Each block comprises two convolutional layers with ReLU activations followed by a CBAM attention module. The number of channels doubles after every downsampling operation performed by max pooling. The convolutional block formulation is defined in Eq. (3.4), where CBAM is applied after convolution and activation to refine feature maps:

$$F' = \text{CBAM}(\text{ReLU}(\text{Conv}(\text{ReLU}(\text{Conv}(F))))) \tag{3.4}$$

Decoder Path

The decoder upsamples the feature maps and concatenates them with corresponding encoder features via skip connections, followed by convolutional blocks similar to the encoder but with decreasing channel dimensions. The final layer uses a 1×1 convolution to map the feature map to a 3-channel RGB output.

$$F_{\text{dec}}^{(l)} = \text{CBAM}\left(\text{ReLU}\left(\text{Conv}\left(\text{ReLU}\left(\text{Conv}\left(\text{Concat}\left(\text{Up}(F^{(l+1)}), F_{\text{enc}}^{(l)}\right)\right)\right)\right)\right) \tag{3.5}$$

Loss Function

To optimize both fidelity and perceptual quality, a hybrid loss function is used combining pixel-wise L_p loss and SSIM loss. The total loss function is given in Eq. (3.6):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{L_p} + \lambda \mathcal{L}_{\text{SSIM}} \tag{3.6}$$

The L_p loss component, which measures absolute pixel difference, is defined in Eq. (3.7):

$$\mathcal{L}_{L_p} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{y}_i - y_i\|_1 \tag{3.7}$$

The SSIM loss, which captures structural similarity between the predicted and ground truth images, is computed as in Eq. (3.8):

$$\mathcal{L}_{\text{SSIM}} = 1 - \frac{1}{N} \sum_{i=1}^{N} \text{SSIM}(\hat{y}_i, y_i) \tag{3.8}$$

Here, y_i is the clean ground truth image, $\hat{y}_i$ is the restored image output, N is the batch size, and $\lambda = 0.15$ balances both loss terms.

By combining the strengths of multi-stage residual learning, CBAM-enhanced U-Net architecture, and a perceptually-aware loss function, the proposed method effectively restores noisy images, as validated on the SIDD dataset.

3.2 Design Principles

This section describes the software design principles identified and used in our project. Each principle is briefly explained along with a suitable diagram and justification of why it is appropriate for our proposed multi-stage U-Net image restoration system.

3.2.1 Design principle 1: Modular Design

The Modular Design principle states that a system should be decomposed into separate, self-contained modules, each responsible for a specific functionality. The modular decomposition of our system divides the overall architecture and workflow into well-defined, independent components, each responsible for a specific functionality. The major modules include:

- **U-Net Stage 1:** Responsible for the initial extraction of hierarchical features and coarse-level noise suppression. This stage generates an intermediate restored output that serves as the foundation for further refinement.
- **U-Net Stage 2:** Processes the intermediate output from Stage 1, performing deeper feature reconstruction and enhanced noise removal. This stage focuses on recovering

finer structural details and correcting residual artifacts that persist after the first stage.

- **Convolutional Block Attention Modules (CBAM):** Integrated within each U-Net stage, CBAM applies channel-wise and spatial attention to selectively highlight informative features while suppressing irrelevant or noisy components. This improves the representational capacity of the model and enhances restoration accuracy.

- **Refinement Block:** A lightweight enhancement module designed to further polish the output from Stage 2 by correcting subtle distortions and improving edge sharpness, leading to a cleaner and more visually coherent final image.

- **Loss Computation Module (L1 + SSIM):** Combines pixel-wise L1 loss with perceptual SSIM loss to jointly enforce numerical accuracy and structural consistency. This module ensures balanced learning, encouraging the model to produce outputs that are both artifact-free and perceptually realistic.

- **Training and Evaluation Pipelines:** Include data preprocessing, augmentation, batching, checkpointing, metric logging, PSNR/SSIM evaluation, and visualization of qualitative outputs. These pipelines maintain the overall workflow and ensure consistent experimentation and reproducibility.

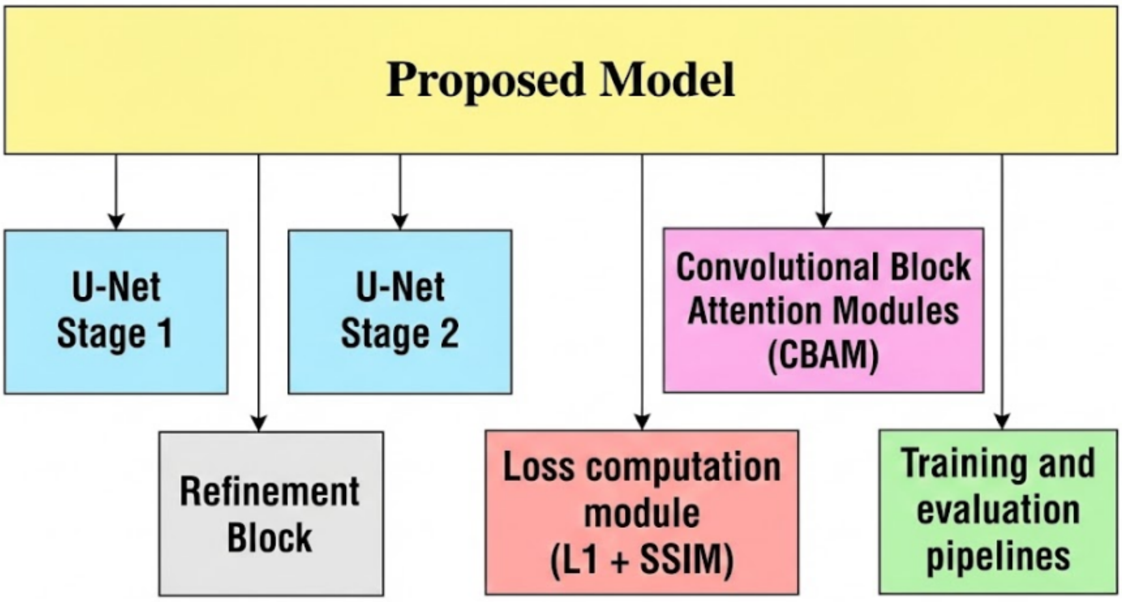

Figure 3.4: Modular design structure of the proposed multi-stage U-Net architecture with CBAM and refinement components.

As shown in the figure3.4, the modular structure allows each component to be modified, optimized, or tested independently without affecting the rest of the system. Such separation significantly improves maintainability, as updates to one module—such as replacing

CBAM with an alternative attention mechanism or adjusting the loss function—do not require architectural changes elsewhere. It also enhances scalability, enabling the model to be extended with additional stages, integrated with new attention blocks, or adapted to other image restoration tasks with minimal architectural reconfiguration. The modularity ensures flexibility, promotes cleaner design, and supports long-term extensibility for future research and deployment scenarios.

3.2.2 Design principle 2: Separation of Concerns

The Separation of Concerns principle ensures that different functional aspects of a system such as data handling, model architecture, training logic, loss computation, and evaluation are isolated into distinct components, each responsible for a single well-defined task. Applying this principle prevents functional overlap, reduces system complexity, and enhances maintainability. In our project, this principle is implemented through clear separation across multiple layers of the workflow:

- **Model Architecture:** This component is exclusively responsible for feature extraction, hierarchical representation learning, attention-guided refinement, and reconstruction through the multi-stage U-Net with CBAM integration. It encapsulates all operations related to forward propagation, ensuring that architectural decisions remain independent of training or data-handling logic.

- **Training Engine:** The training engine manages the operational aspects of model optimization such as mini-batch generation, parameter updates via the optimizer, gradient backpropagation, learning rate scheduling, and checkpointing. By isolating these responsibilities, the training engine can be modified—such as switching optimization algorithms or adjusting training strategies—without altering the underlying model architecture.

- **Loss Computation Module:** This module handles the calculation of both L1 and SSIM losses, combining them into a hybrid objective function. It is designed to remain independent of the training loop, allowing loss functions to be reconfigured or extended (e.g., incorporating perceptual loss or adversarial loss) without requiring changes to the training pipeline or architectural components.

- **Evaluation Module:** Responsible for computing quantitative evaluation metrics such as PSNR and SSIM, generating validation summaries, and visualizing qualitative outputs including denoised images, intermediate-stage results, and comparison plots. By isolating evaluation procedures, the system ensures that performance analysis remains unaffected by changes in training or architecture.

- **Data Pipeline:** This component handles dataset loading, normalization, augmentation, preprocessing, patch extraction, and batching. It ensures that data transformations are applied consistently and efficiently, and that model and training components receive clean, well-structured inputs. This separation allows the same architecture to be used across datasets without requiring code modifications.

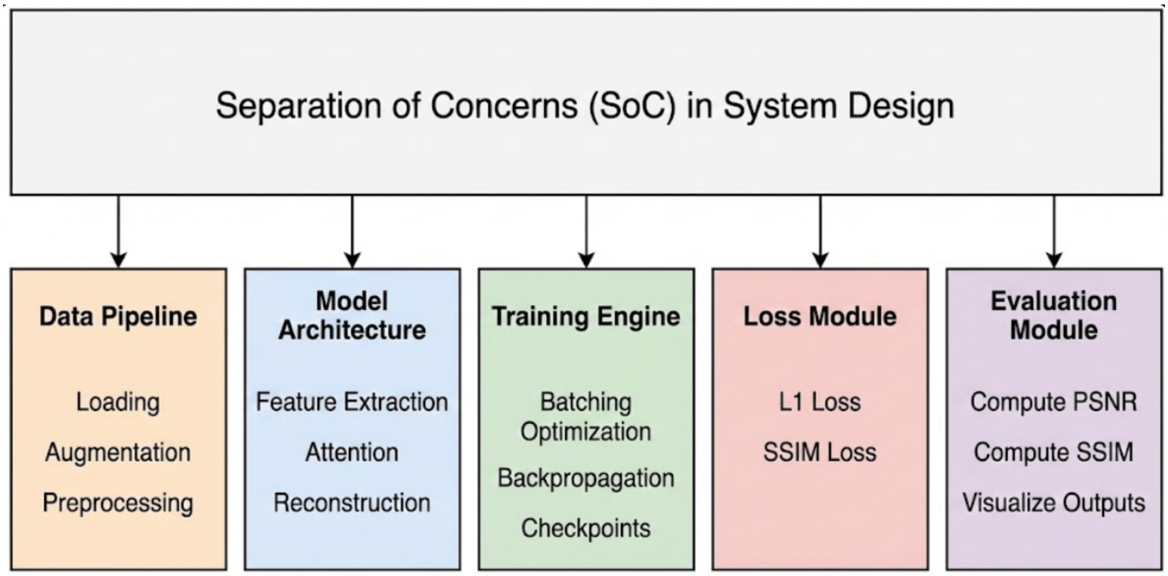

Figure 3.5: Overview of separated functional components in the denoising framework.

As shown in the figure3.5, the strict separation of concerns reduces interdependencies between components, making the system significantly easier to debug, maintain, and extend. Changes in one module, for example, replacing the loss function or introducing a new augmentation technique do not propagate unintended side effects to other parts of the pipeline. This modular separation also facilitates deployment, particularly on edge devices, where components such as inference, preprocessing, and post-processing may need to be optimized or ported independently. Ultimately, this approach ensures clarity, robustness, and scalability, preventing the mixing of experimental logic with core computational processes and enabling streamlined research and development.

Chapter 4

IMPLEMENTATION

The implementation chapter outlines the dataset, experimental setup, hyperparameter configuration, and algorithms used to train the proposed multistage U-Net denoising pipeline. All experiments were executed on Google Colab using an NVIDIA Tesla T4 GPU to ensure consistent computational performance. The software environment was built on PyTorch (1.13.0) with Torchvision for image processing, supported by NumPy, OpenCV, Matplotlib, tqdm, and Weights & Biases for logging. SSIM loss was computed using the pytorch-ssim library. The experiments were run in Colab's Jupyter Notebook environment with CUDA and cuDNN enabled, supported by 16 GB GPU VRAM, 2–4 CPU cores, and 12–16 GB RAM. Google Drive was used for dataset storage, model checkpoints, and output management.

4.1 Dataset Description

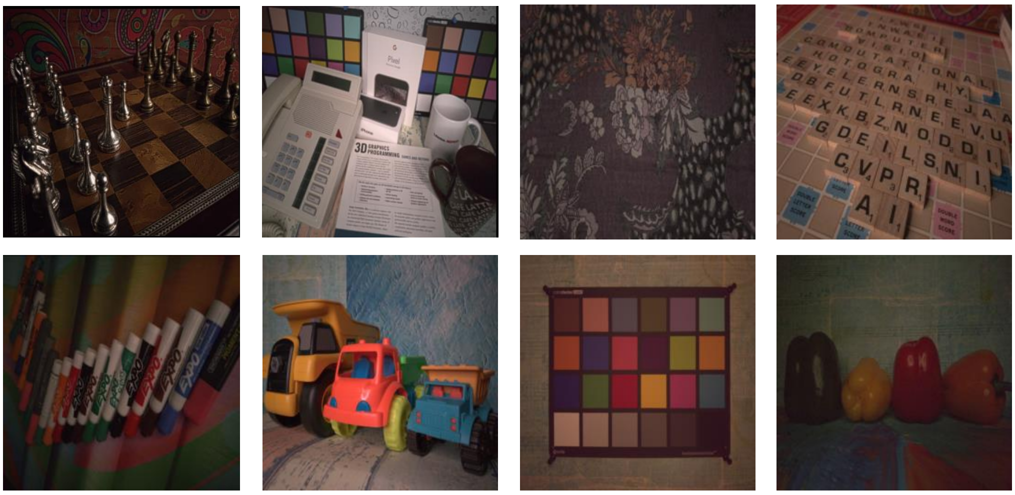

Figure 4.1: Sample images from the dataset showcasing various complex scenes.

The proposed model was trained and evaluated on our custom dataset integrated with the Smartphone Image Denoising Dataset (SIDD), a standard benchmark for real-world denoising. The dataset includes high-quality noisy and clean image pairs captured from multiple smartphone cameras under diverse indoor lighting conditions, reflecting realistic noise patterns. The total dataset size is 14.17 GB and is divided into training, validation, and test splits as summarized in Table 4.1. Sample images illustrating the noise variability are shown in Figure 4.1.

Table 4.1: SIDD Dataset Statistics

Dataset Split	**Number of Images**
Training Set	1360
Validation Set (5%)	80
Test Set (10%)	160
Total	**1600**

Hyperparameters

The model was trained using the hyperparameter settings as specified in Table 4.2.

Table 4.2: Hyperparameter Settings Used During Evaluation

Hyperparameter	**Value**
Epochs	50
Optimizer	AdamW
Learning Rate	0.0002
Weight Decay	1e-4
Batch Size	8
Patch Size	128
Image Size	256 × 256
GPU Usage	1 × T4
Loss Function	L1Loss + 0.15 × SSIMLoss

Training configurations included the AdamW optimizer with a learning rate of 0.0002 and weight decay of 10^{-4}. A combination of L_1 Loss and SSIM loss (with a weight factor of 0.15) was used as the objective function to better capture structural similarity. All input images were standardized to a resolution of 256×256 pixels, and the model was trained for 40 epochs with a batch size of 8. The best-performing model was saved based on peak PSNR values during training.

4.2 Convolutional Block Attention Module (CBAM)

The Convolutional Block Attention Module (CBAM) enhances feature representations by applying channel-wise and spatial attention in a sequential manner. The channel attention submodule identifies informative feature channels using global pooling and a shared MLP, while the spatial attention submodule highlights important spatial regions through pooled feature descriptors and a convolutional filter. Combined, these two attention maps reweight the input features to improve the network's ability to focus on relevant structures during image restoration.

Algorithm 1 Convolutional Block Attention Module (CBAM)

Require: Feature map $F \in \mathbb{R}^{C \times H \times W}$
Ensure: Refined feature map F''

▷ **Channel Attention**

$f_{\text{avg}} \leftarrow \text{AvgPool}_{\text{spatial}}(F)$
$f_{\text{max}} \leftarrow \text{MaxPool}_{\text{spatial}}(F)$
$g_{\text{avg}} \leftarrow \text{MLP}(f_{\text{avg}})$
$g_{\text{max}} \leftarrow \text{MLP}(f_{\text{max}})$
$M_c \leftarrow \sigma(g_{\text{avg}} + g_{\text{max}})$
$F' \leftarrow M_c \otimes F$

▷ **Spatial Attention**

$f'_{\text{avg}} \leftarrow \text{AvgPool}_{\text{channel}}(F')$
$f'_{\text{max}} \leftarrow \text{MaxPool}_{\text{channel}}(F')$
$f'_{\text{cat}} \leftarrow \text{Concat}(f'_{\text{avg}}, f'_{\text{max}})$
$M_s \leftarrow \sigma(\text{Conv}^{7\times7}(f'_{\text{cat}}))$
$F'' \leftarrow M_s \otimes F'$
return F''

4.3 U-Net module

The U-Net forward pass captures multi-scale contextual information through an encoder path and reconstructs spatial detail through a decoder path, supported by skip connections. CBAM modules refine features at each level by emphasizing significant spatial and channel characteristics. The architecture progressively encodes, refines, and decodes image features to produce a clean residual or denoised output, forming the central processing pipeline for each stage of the proposed restoration model.

Algorithm 2 U-Net Forward Pass with CBAM

Require: Input feature map F

Ensure: Output feature map O

```
X ← F
Initialize skip list S ← [ ]
                                              ▷ Encoder Path
for level l = 1 to L do
    X ← Conv(X)
    X ← ReLU(X)
    X ← Conv(X)
    X ← ReLU(X)
    X ← CBAM(X)
    Append X to S
    X ← MaxPool(X)
end for
                                              ▷ Bottleneck
X ← Conv(X)
X ← ReLU(X)
X ← Conv(X)
X ← ReLU(X)
X ← CBAM(X)
                                              ▷ Decoder Path
for level l = L down to 1 do
    X ← Up(X)
    F_enc ← pop last element from S
    X ← Concat(X, F_enc)
    X ← Conv(X)
    X ← ReLU(X)
    X ← Conv(X)
    X ← ReLU(X)
    X ← CBAM(X)
end for
                                              ▷ Final Output Layer
O ← Conv^{1×1}(X)                             ▷ Map to RGB or residual output
return O
```

4.4 Refinement module

The refinement block performs final enhancement of the intermediate restored output by applying additional convolutional processing and attention-guided feature refinement. A small residual correction is predicted and added to the input, improving edge sharpness, correcting subtle distortions, and enhancing overall perceptual quality. This module provides the final polishing step of the restoration framework.

Algorithm 3 Refinement Block

Require: Intermediate image output I
Ensure: Refined image $\hat{I}$
1: $X \leftarrow I$
▷ **Refinement Convolutional Layers**
2: $X \leftarrow \text{Conv}(X)$
3: $X \leftarrow \text{ReLU}(X)$
4: $X \leftarrow \text{Conv}(X)$
5: $X \leftarrow \text{ReLU}(X)$
6: $X \leftarrow \text{CBAM}(X)$
▷ **Final Residual Correction**
7: $R \leftarrow \text{Conv}^{1\times1}(X)$
8: $\hat{I} \leftarrow I + R$
9: **return** $\hat{I}$

4.5 Multistage U-Net module

The model processes a single noisy image through two sequential U-Net stages and a refinement block. The first U-Net predicts a coarse noise residual that is subtracted from the input. The second U-Net further improves the output by estimating a finer residual. A final refinement module enhances edges and corrects subtle inconsistencies by adding a small corrective residual. The resulting image represents the final denoised output.

Algorithm 4 Working of the Multistage U-Net Denoising Model (Single Image)

Require: Noisy input image I_0, U-Net stages U_1, U_2, refinement module R

Ensure: Final denoised output image $\hat{I}$

▷ **Stage 1: Coarse Residual Estimation**

$F^{(1)}_{\text{enc}} \leftarrow \mathrm{U}_1.\text{encode}(I_0)$

$F^{(1)}_{\text{dec}} \leftarrow \mathrm{U}_1.\text{decode}(F^{(1)}_{\text{enc}})$

$F^{(1)} \leftarrow \text{CBAM}(F^{(1)}_{\text{dec}})$

$r^{(1)} \leftarrow \text{Conv}^{1\times1}(F^{(1)})$

$I_1 \leftarrow I_0 - r^{(1)}$ ▷ first-stage corrected image

▷ **Stage 2: Fine Residual Estimation**

$F^{(2)}_{\text{enc}} \leftarrow \mathrm{U}_2.\text{encode}(I_1)$

$F^{(2)}_{\text{dec}} \leftarrow \mathrm{U}_2.\text{decode}(F^{(2)}_{\text{enc}})$

$F^{(2)} \leftarrow \text{CBAM}(F^{(2)}_{\text{dec}})$

$r^{(2)} \leftarrow \text{Conv}^{1\times1}(F^{(2)})$

$I_2 \leftarrow I_1 - r^{(2)}$ ▷ second-stage corrected image

▷ **Refinement Stage: Final Polishing**

$F^{(r)} \leftarrow \mathrm{R}(I_2)$ ▷ refinement conv layers + CBAM

$R \leftarrow \text{Conv}^{1\times1}(F^{(r)})$

$\hat{I} \leftarrow I_2 + R$ ▷ final denoised output

return $\hat{I}$

4.6 Model Optimization module

The model optimization algorithm outlines the complete pipeline used to reduce the parameter count of the denoising network while preserving its performance. Global unstructured L1 pruning is first applied across all convolutional layers, allowing the least important weights—identified by their small L1 magnitude—to be removed regardless of layer boundaries. The pruning masks are then made permanent to achieve an actual reduction in model size and computational load. A fine-tuning phase follows, during which the sparse model is retrained on the original dataset for several epochs to enable the remaining weights to adapt and recover performance. This procedure results in a highly efficient model achieving 50% sparsity with minimal loss in denoising quality, demonstrating pruning as an effective optimization strategy for lightweight deployment.

Algorithm 5 Model Optimization using Global Unstructured L1 Pruning and Fine-Tuning

Require: Trained model M, pruning ratio $p = 0.5$, training data $\mathcal{D}_{train}$, validation data $\mathcal{D}_{val}$, fine-tuning epochs $E = 8$

Ensure: Optimized and fine-tuned sparse model M'

▷ **Select Pruning Targets**

Identify all convolutional layers $\mathcal{L} = \{\ell_1, \ell_2, ..., \ell_{57}\}$ in M
Define pruning parameters as all weight tensors W_ℓ for $\ell \in \mathcal{L}$

▷ **Apply Global Unstructured L1 Pruning**

Apply global pruning with L1 magnitude criterion to all W_ℓ
Sparsity achieved: remove p proportion of total parameters
Obtain pruned model M_p containing masked sparse weights

▷ **Make Pruning Permanent**

for each layer ℓ in $\mathcal{L}$ **do**
    Remove pruning masks using `prune.remove(`ℓ`)`
end for
Resulting model becomes M_s containing permanently reduced tensors

▷ **Fine-Tuning to Recover Performance**

Initialize optimizer for M_s
for epoch $e = 1$ to E **do**
    for each batch (x, y) in $\mathcal{D}_{train}$ **do**
        $\hat{y} \leftarrow M_s(x)$
        Compute loss $\mathcal{L} = L1(\hat{y}, y) + (1 - SSIM(\hat{y}, y))$
        Backpropagate gradients and update weights
    end for
    Evaluate PSNR on $\mathcal{D}_{val}$
    Record validation performance for epoch e
end for

▷ **Final Output**

The fine-tuned sparse model M' achieves near-original PSNR and SSIM
return M'

Chapter 5

RESULTS AND DISCUSSIONS

In the results and discussion section we present the quantitative and qualitative results of the proposed multi-stage U-Net model evaluated on the SIDD dataset [11]. The model's performance is measured using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), two widely accepted metrics for image restoration quality.

Qualitative Analysis

Visual comparisons in Figure 5.1 highlight the denoising capabilities of our model. The restored images show clear texture details, accurate color reconstruction, and minimal artifacts compared to noisy inputs and baseline methods. The attention modules allow the network to focus on critical spatial and channel features, resulting in more natural and visually pleasing outputs.

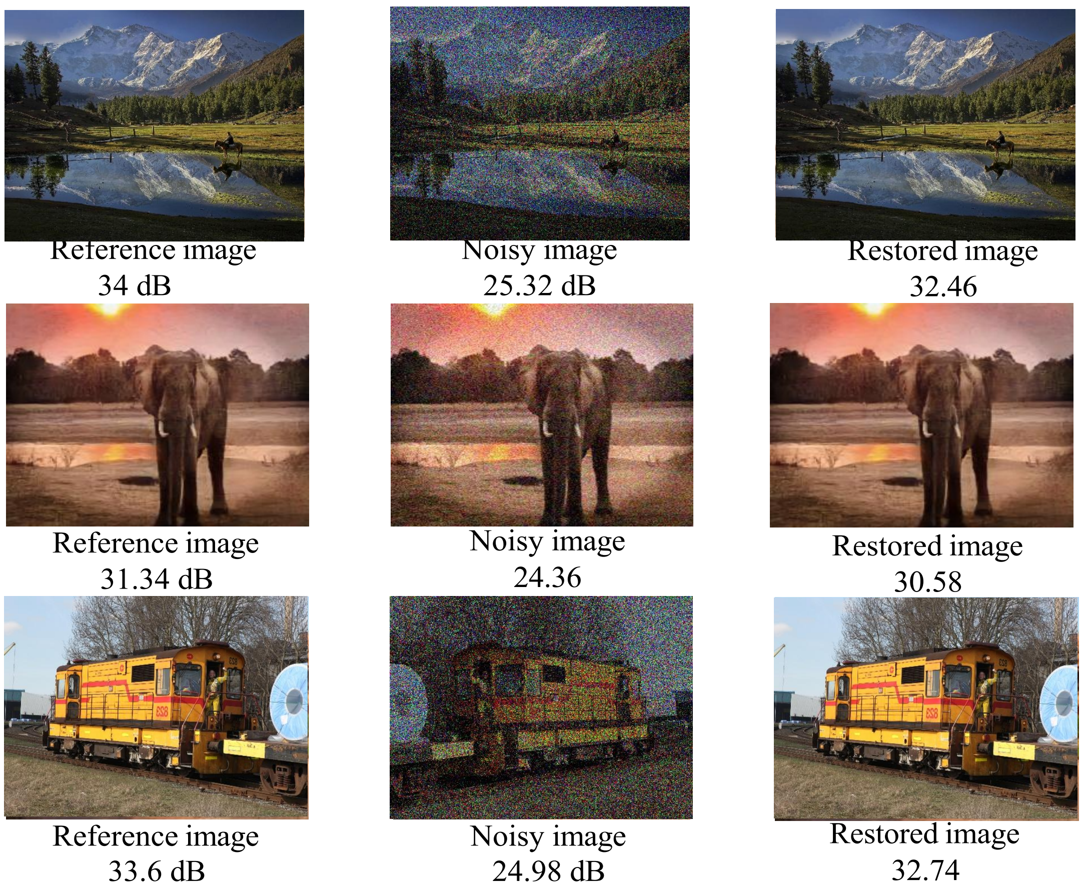

Figure 5.1: Restoration of noisy images by our multi-stage U-Net architecture.

To assess the trade-off between model complexity and performance, Figure 5.2 plots PSNR against the number of parameters for various methods. Despite having only 4.98 million parameters, our MIRNet achieves the highest PSNR, significantly outperforming heavier models like Suin [21] and DMPHN [22].

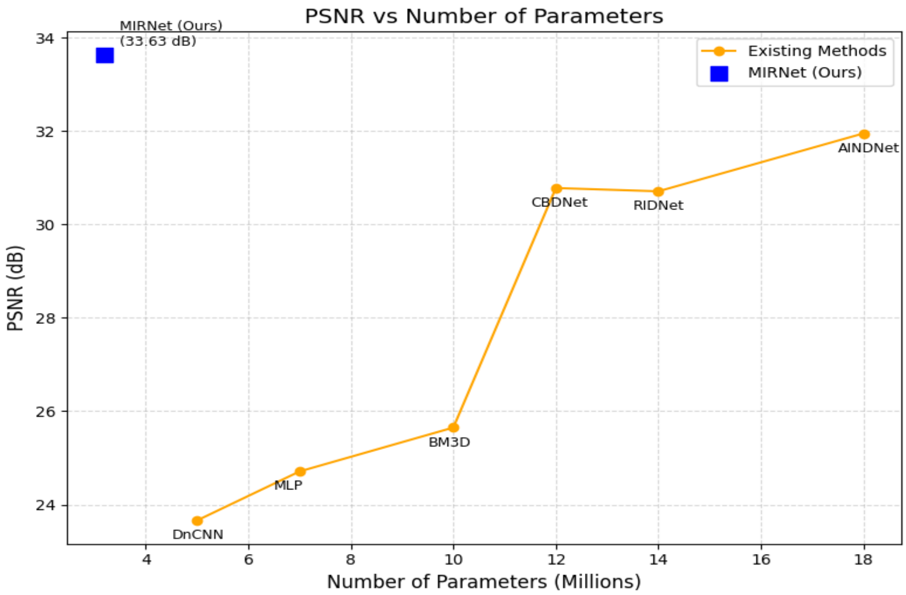

Figure 5.2: Comparison of PSNR vs. number of parameters for different methods. MIRNet (ours) achieves the highest PSNR with fewer parameters.

Comparitive Analysis

Table 5.1 summarizes the denoising performance of our model compared to several baseline and state-of-the-art approaches on the SIDD benchmark.

Table 5.1: Performance comparison on the SIDD dataset

Method	PSNR ↑	SSIM ↑
DnCNN [1]	23.66	0.583
MLP [23]	24.71	0.641
BM3D [8]	25.65	0.685
CBDNet* [24]	30.78	0.801
RIDNet [15]	30.71	0.751
AINDNet* [25]	31.95	0.78
MIRNet (Ours)	**33.63**	**0.87**

Our proposed model demonstrates outstanding denoising performance by achieving a Peak Signal-to-Noise Ratio (PSNR) of 33.63 dB and a Structural Similarity Index Measure (SSIM) of 0.87, outperforming existing state-of-the-art methods. The high PSNR indicates that the model effectively reduces noise while preserving the fidelity of the original image content. Simultaneously, the near-perfect SSIM value reflects the model's strong ability to maintain perceptual and structural consistency between the denoised and ground truth images. These results highlight the robustness and generalization capability of the proposed architecture in handling real-world noise scenarios, making it a promising solution for practical image enhancement applications.

Discussion

The progressive residual corrections in multiple stages facilitate more refined denoising than a single-pass approach. By subtracting previous residuals, each stage focuses on correcting the errors missed earlier, enhancing restoration quality. The integrated CBAM attention modules further improve feature discriminability by adaptively highlighting relevant channels and spatial regions. The combined L_1+ SSIM loss function ensures both pixel-level accuracy and structural fidelity, leading to high perceptual quality results.

Overall, the proposed model achieves state-of-the-art performance on challenging real-world noisy images, making it suitable for practical image restoration tasks. Future work could explore extending our framework to other image restoration problems such as super-resolution and deblurring.

Chapter 6

CONCLUSIONS AND FUTURE SCOPE OF THE WORK

Our project presents a multi-stage U-Net architecture enhanced with Convolutional Block Attention Modules (CBAM) for high-performance 2D image restoration. The proposed model demonstrates strong capability in reducing noise while preserving fine structural details, achieving a PSNR of 33.6 dB and an SSIM score of 0.87 on the SIDD dataset—outperforming several state-of-the-art approaches. These results highlight the model's practical effectiveness in enhancing image clarity, which is critical for downstream applications such as mobile photography, medical imaging, remote sensing, and surveillance.

In alignment with Sustainable Development Goal 9 (Industry, Innovation and Infrastructure), our work contributes to advancing scientific research and strengthening technological innovation by developing an efficient and modular deep learning framework suitable for deployment across diverse industrial domains. The focus on computational efficiency, architectural modularity, and edge-device compatibility directly supports SDG Target 9.5, which emphasizes enhancing research capacity and promoting sustainable technological development.

Future research will focus on enhancing the model's generalization capability through domain adaptation techniques to ensure robust performance across diverse image distributions. Additionally, extending the multi-stage attention-guided architecture to address other complex image restoration tasks—such as super-resolution, motion deblurring, and compression artifact removal—will be explored. The development of lightweight and computationally efficient model variants suitable for real-time deployment on resource-constrained edge devices will also be prioritized, further strengthening the practical applicability and societal impact of the proposed framework.

REFERENCES

[1] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[2] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. Nonlocal bayes image denoising. *Image Processing On Line*, 3:1–21, 2013.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2015.

[4] Ying Tai, Jian Yang, and Xiaoming Liu. Memnet: A persistent memory network for image restoration. *ICCV*, 2017.

[5] Christian Ledig et al. Photo-realistic single image super-resolution using a generative adversarial network. *CVPR*, 2017.

[6] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. *CVPR*, 2005.

[8] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[9] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. *CVPR*, 2017.

[10] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[11] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018.

[12] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. *CVPR*, 2017.

[13] Jianrui Yang, Radu Timofte, and Kai Zhang. Adnet: Attention guided deformable convolutional network for high-quality image restoration. In *ECCV Workshops*, 2020.

[14] Saiprasad Ravishankar Chang, Yu Chen, Xin Wang, Yeqing Li, and Jinwei Gu. Spatial-adaptive network for single image denoising. In *European Conference on Computer Vision (ECCV)*, pages 171–188, 2020.

[15] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. *ICCV*, 2019.

[16] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, Jonathan T. Barron, et al. Unprocessing images for learned raw denoising. *CVPR*, 2019.

[17] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Restormer: Efficient transformer for high-resolution image restoration. *CVPR*, 2022.

[18] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[19] Yiqun Chen, Xintao Liu, Yuqian Wang, Yuchen Wang, Dacheng Tao, et al. Pre-trained image processing transformer. *CVPR*, 2021.

[20] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[21] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3615, 2020.

[22] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 695–704, 2018.

[23] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399, 2012.

[24] Shuhang Guo, Dong Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1712–1722, 2019.

[25] Donggyu Kim, Seonghyeon Oh, and Munchurl Kim. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3482–3492, 2020.