# Multi-patch Hierarchical Network with Non-local Information for Real World Image Denoising

by

**KRISHNA SAVALIYA**
**202011050**

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

**DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY**

May, 2022

## Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Krishna Savaliya

## Certificate

This is to certify that the thesis work entitled MULTI-PATCH HIERARCHICAL NETWORK WITH NON-LOCAL INFORMATION FOR REAL WORLD IMAGE DENOISING has been carried out by KRISHNA SAVALIYA for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Prof. Srimanta Mandal
Thesis Supervisor

# Acknowledgments

I am deeply grateful to my mentor, Prof. Srimanta Mandal. Success is not a one night thing, but it is the result of hard work, learning from failures, and consistent effort and I would like to thank my supervisor, Prof. Srimanta Mandal, for instilling these qualities in me over the last year. He was always there to help me whenever I was stuck and could not see the road ahead. His motivation and constant support has motivated me to do the quality research in my masters. Also, I would like to give deep sense to all faculty members for valuable suggestions and directions during presentations. I am thankful to the enthusiastic support given to us by the college in providing a competitive and supportive environment that allowed me to do great work in research.

Last but not least; I would like to acknowledge the unquestioning and tireless support from my friends and family.

# Contents

# Abstract

In the domain of image denoising, there has been significant and rapid development recently. Many prior noise-modeling based and deep-learning based algorithms have shown outstanding results in denoising. However, the networks used by the state-of-the-art methods are very deep and complex. We propose a simple yet effective Deep Multi-patch Hierarchical Network that uses less memory and has fewer network parameters. In this network, multiple features of noisy image patches from different spatial regions are combined using a fine-to-coarse hierarchical representation to get a clean image. While deep neural networks have had great success in denoising images using additive white Gaussian noise (AWGN), their performance on real-world noisy images is weak. This is because their trained models are likely to overfit the simplified AWGN model, which differs significantly from the complex real-world noise model. Hence we trained our model by real-world noisy data to generalize the ability of the denoise network. In the encoder section of the network, we also added a non-local module to extract dependencies between long-distant pixels in the image, which enhanced PSNR to 0.25 dB. Additionaly, The parallel connection of channel attention (CA) and pixel attention (PA) is added into the decoder to further enhance the performance. Different channels and pixels contain different levels of important information, and attention can give more weight to relevant information so that the network can learn more useful information. This resulted in a PSNR increment of 0.17 dB. When compared to most deep learning approaches, our architecture shows competitive results while using fewer network parameters.

**Keywords:** *real-world noise, image denoising, non-local, channel-pixel attention*

# List of Principal Symbols and Acronyms

| | |
|---|---|
| **AWGN** | Additive White Gaussian Noise |
| **CA** | Channel Attention |
| **DMPHN** | Deep Multi-Patch Hierarchical Network |
| **PA** | Pixel Attention |
| **PSNR** | Peak Signal-to-Noise Ratio |
| **SPM** | Spatial Pyramid Matching |
| **SSIM** | Structural SIMilarity |
| **TV** | Total Variation |

# List of Tables

# List of Figures

# CHAPTER 1
# Introduction

Almost all the imaging equipment induces noise to a certain extent during image acquisition and transmission. As a result, the quality of captured or received images often becomes low. Image denoising is a significant and fundamental task in low-level vision due to its wide range of applications in different domains, such as medical image denoising [15], remote-sensing image denoising [41], satellite image denoising [33], and compression noise removal [34]. Even the higher-level vision tasks such as object detection [27] and recognition [12] require the image to be as clean as possible.

The main objective of image denoising is to remove or correct noise in an image, either for aesthetic purposes or to improve the performance of several other down-stream tasks. Over decades of study, several promising approaches [10] for the removal of additive white Gaussian noise (AWGN) have been discovered, and near-optimal results have been achieved [19, 29]. On the contrary, image noise in real-world camera systems arises from a number of sources (e.g., thermal noise, dark current noise, shot noise, and so on). Further, an image gets impacted by the in-camera processing (ISP) pipeline (e.g., demosaicing, compression, etc). Hence, the distribution of real-world noise is different than the AWGN. To reduce the effect of real-world noise from an image, we propose a deep learning technique based on a hierarchical framework.

We consider three stages for denoising an image. The output features of the third stage are fed to the second, and the output of the second stage is fed to the first to produce the final denoised result. At the last stage, we segment the input image into four patches, thus focusing on local details, whereas we fed the entire image into the first stage, thus looking at the global aspect. Hence, in our architecture, we consider local-to-global information of an image while denoising in our architecture.

Each stage of the network consists of an encoder and a decoder. The encoder is designed with a few non-local blocks to explore the non-local patch similarity.

The decoder is equipped with an attention mechanism, where channel and pixel attention modules are used in parallel. Channel attention assigns weights to each channel according to the relevance, whereas pixel attention focuses on each pixel's information.

## 1.1 Objectives

The objectives of the work are

- To denoise an image using a deep learning framework when the noise can have any kind of distribution.
- To incorporate local-to-global information while denoising an image.
- To utilize similar information in an image, as has been done in many classical denoising methods.
- To assign weights to each channel and pixel according to their relevance.

## 1.2 Contribution

The main objective of our network is to improve performance by combining features of multiple image patches from different spatial regions of the image. Due to residual connections in our model, our encoder and decoder parameters are relatively less, which enables faster denoising inference. So the main idea here is to make the lower-level network focus on retrieving local feature details from the finer grid to generate residual information for the upper-level network. The global information can be obtained from both the finer and coarser grid by concatenating convolutional features. Furthermore, the network is lightweight so that it can be used on mobile GPUs as well.

Also, existing deep learning algorithms do not make use of attention and non-local self-similarity of natural images. To use those similarities, we included a non-local module [39] in the encoder section, which calculates the output at a position as a weighted sum of the features at all positions. Thus, it can detect the long-distance dependencies among distant pixels in an image.

In the decoder, we added a parallel combination of channel and pixel attention since channel attention (CA) may bring out channel-wise relevant information while pixel attention (PA) propagates pixel-level attention. [43] has made use of their serial connections. However, as compared to serial connections, their

parallel connection might better complement each other. In the case of serial connections, any failure in channel attention can disturb the pixel attention process also. Furthermore, in the output of channel attention, the pixel-level information may get affected. Therefore , pixel attention on the output of channel attention will be ineffective since the PA will be unable to assess the unaffected pixel-level data. This problem can be addressed by attaching CA and PA in parallel.
Following are the key aspects of our method:

- Denoising performance improved by combining features of multiple image patches from different spatial regions of the image while considering nonlocality of multiple patches.
- Due to residual connections in the proposed model, our encoder and decoder parameters are relatively low, enabling faster denoising inference.
- We added a parallel combination of channel and pixel attention in the decoder to get channel-wise and pixel-level relevant information from an image.
- In the case of serial connections, any failure in channel attention can disturb the pixel attention process, so a parallel connection of CA and PA is used.

## 1.3 Organization of Thesis

The rest of the thesis is arranged as follows. Chapter 2 contains a literature survey of existing denoising techniques. Chapter 3 explains the proposed model and includes a detailed explanation of its components like non-local and attention modules. Chapter 4 consists of the experimental results of the proposed method, comparing different existing approaches of denoising, followed by other additional experiments. Chapter 5 gives the conclusion of the work and discusses the future scope.

CHAPTER 2

# Literature survey

## 2.1 Classical image denoising methods

In [5], authors proposed a novel sparse representation based denoising method of block-matching and 3D filtering (BM3D). This method is mainly focused on 3 components 1) 3-D transformation of a group 2) reduction of the transform spectrum 3) inverse 3D transformation. Buades et al. [4] proposed a novel algorithm named the non-local means (NLM), this algorithm is based on a non-local averaging of distant image pixels. This non-local method was transformed in a neural network block by [39], which showed significant improvement in classification and segmentation tasks. Zhang et al. [10] proposed WNNM, which is a model-based technique that provides a low-rank recovery method for estimating the denoised image using the rank minimization. The WNNM model presumes that the underlying data may be represented by a low-dimensional linear structure or subspace. It seeks and stacks non-local identical patches for each underlying patch, then uses a convex optimization approach to build a low rank matrix by minimizing an objective function.

The review of various denoising methods reveals that traditional model-based approaches are flexible in dealing with noise at various levels, but they suffer from a number of limitations such as a time-consuming optimization techniques, reliance on appropriate prior noise model and patches with non-local self-similarity.

## 2.2 CNN based methods

The use of convolutional neural networks (CNNs) for image denoising has risen rapidly in recent years [49]. When compared to prior model-based techniques, CNNs deliver faster inference and better performance. The number of factors in CNN designs has increased in recent years, raising the architecture’s complexity. Most of these models have improved their performance over time [42]. A variety of models, on the other hand, require prior knowledge [36] about the type and degree of the noise, or an approximation of it, in order to get the best results. Modern neural networks of image denoising [38] have been shown to function effectively with lesser parameters. Although these models are trained for a specific noise level and need the construction of a model instance for each noise level. Zuo et al. [46] proposes a CNN-based denoising architecture (DnCNN) that outperforms all classical non-CNN-based algorithms on AWGN noise. They modified simple CNN by adding batch normalization (BN) [14] and residual learning [28] into a model to make the network stable and faster. It is an end-to-end denoising system that does not involve any additional noise information. But, the model's performance is sub-standard while using real noise and it also smoothes out the edges in the images. Zhang et al. [47] have introduced the FFDNet (fast and flexible denoising network), which can handle a broad variety of noise levels with a single denoising model. They claim that traditional image denoising models are trained to eliminate noise from images with a certain noise level, forcing them to train different models for various noise levels [46, 35]. To deal with multiple noise levels, FFDNet trains its model using training images with different noise levels. However, FFDNet has a drawback when removing noise from images with an unknown noise level since it needs the image noise level as input data. Lefkimmiatis presented another technique to overcome the constraint of traditional denoising networks, which need a model for each noise level. He proposed the universal denoising network (UDN) [18], covering half of the noise level range with a single network. Although, UDN requires the image's noise level as input. Liu et al. [21] introduced multi-level wavelet CNN (MWCNN), a model that incorporates a modified UNet [30] architecture with the wavelet transform [22]. While they increased the receptive field of their model while lowering its computational cost, they were unable to overcome the drawback that their model is only useful for a single noise level. Furthermore, their usage of the wavelet transform might degrade performance by forcing their network to utilize wavelet transform feature information.

## 2.3 SPM based methods

Svetlana et al. [17] suggest Spatial Pyramid Matching (SPM) for image classification and object recognition. It splits images into coarse-to-fine grids and computes feature histograms. It combines many image patches to enhance the scene recognition performance. Inspired by this idea, SPM-based architecture is also used in image deblurring [44] and image dehazing [6]. We use it for image denoising. Using an SPM-like model has two major advantages. The input of multiple levels has an identical spatial resolution. Therefore, residual-like learning requires a lower filter size and faster inference performance. Second, by utilizing an SPM-like model, more training data could be exposed to the finest scale, which leads to more patches, resulting in improved performance.

Existing multi-scale-based techniques require a considerable amount of time and memory. To solve this limitation [7] proposed DMPHN and DMSHN which has multi-level architecture. Also, this method has a very short time to execute as compared to other multi-scale-based methods. DMPHN contains 3 levels with different scales 1,2 and 4 respectively. The model integrates local features generated from a finer to a coarser resolution level. The encoder-decoder architecture is used in each level of the model. This Architecture's encoder includes 15 convolution layers and six residual connections. The architectures of the encoder and decoder are identical, except for two convolutions that have been replaced by Deconvolution layers.

## 2.4 Image denoising datasets

Based on the source of the given noisy images within the dataset, image denoising datasets can be classified as: synthetic noisy image dataset [48] and real-world noisy image dataset [40]. To create a synthetic image dataset, first collect ground truth noise-free images by down sampling a high-resolution image or by post-process a low-ISO image [26]; then add synthetic noise based on statistic models (such as the Gaussian model or Poisson-Gaussian mixture model [9]) to obtain noisy images. A method to create a real-world noisy dataset is: 1) Gathering many real-world noisy images in a short period of time to ensure the minimum content change in the image, such as scene brightness changes or scene object motions; 2) merging all these images to obtain ground truth "noise-free" image. The real-world noisy dataset is closer to the actual image data handled in practical applications than the synthetic noise dataset. Therefore, we consider denoising on the real-world noisy dataset.

Many datasets with real-world noisy images were proposed. Anaya et al. proposed the RENOIR dataset [2], which comprises pairs of low or high ISO images. The low-ISO photos still have considerable noise, and the dataset fails to capture precise spatial alignment. Plotz et al. [26] proposed Darmstadt Noise Dataset (DND). It performs post-processing to 1) spatially align low-ISO images with their high-ISO equivalents and 2) eliminate intensity variations caused by atmospheric light or light flickering. Although majority of DND images contain minimal levels of noise and lighting settings. Hence, there are only a few instances of high noise levels or low lighting conditions. Abdelhamed et al. [1] proposed the SIDD dataset, which addressed issues with previous datasets such as spatial misalignment and clipped intensities because of low-light scenarios. We used SIDD dataset to analyze the denoising performance of our proposed approach. Because of the wide range of noise levels and lighting settings, SIDD is a good choice for benchmarking models' denoising performance.

# CHAPTER 3
# Multi-patch Hierarchical Network with Non-local Information

## 3.1 Multi-patch Hierarchy

The multi-patch hierarchy schemes are referred to as structures that convey information through levels of the hierarchy and aggregation of several image patches. The multi-patch hierarchy comprises numerous low-to-high layers, using image blocks of different sizes as input to each level. Information from different stages is combined through residual connections between features and images. The scheme is illustrated in Figure 3.1.

More image blocks are available on the lowest level by processing image patches individually, which is equivalent to increasing training data. Therefore, the input image's local features can be maintained and transferred up the hierarchy through residual connections. Hence, each hierarchy level can focus on various intensities of noise. Unlike prior methods that used Gaussian pyramids as input, the multi-patch hierarchy approach takes advantage of an image's inherent spatial scale patches, which significantly reduces the feature extraction time. In summary, the new algorithm is more focused on local details of images and speeds up feature extraction. Higher-level image interpretation is facilitated by lower-level features and images conveyed through residual connections in the multi-patch architecture. The network generally takes input in the form of multi-patch hierarchical images to take advantage of higher and middle-level information while maintaining low-level information. The encoder and decoder outputs of the lower level are sent up to the upper level by residual connections. An image is gradually recovered from each level.

## 3.2 Skip connection

Noroozi et al. [25] designed a skip connection across input and output to decrease the effort of image restoration while maintaining image consistency. In general, we establish a skip link between the generated image and the input image to confirm that the output image matches the original. However, we use a skip connection between the input image and semi-processed image restored on each level, to accumulate details of all levels on the final level. This concatenation propagates not just the information lost because of down-sampling, but also some interfering information. The high-level features become more ample as data propagates across the network, and noisy deviation in the image is decreased.

## 3.3 Network Architecture

We use Deep Multi-patch Hierarchical Network (DMPHN), which is basically designed for single image deblurring [45]. In this research, we employ the (1-2-4) form of DMPHN. To be thorough, we will go over the architecture in the following sections. The DMPHN architecture is multi-level. Each level has an encoder-decoder pair. Each level operates on a different number of patches. Correspondingly, the number of patches utilized in DMPHN(1-2-4) is 1,2 and 4 from top to bottom levels. The highest level (level-1) takes into account only one patch per image. The image is divided into two sections vertically in the following level (level-2). At the bottom-most level (level-3) the patches are further divided horizontally, giving a total of 4 patches.

Consider the input noisy image $I^N$. $I^N_{i,j}$ denotes the $j$-th patch in the $i$-th level. $I^N$ is not separated into patches in level 1. $I^N$ is separated vertically in $I^N_{2,1}$ and $I^N_{2,2}$ on level 2. further, $I^N_{2,1}$ and $I^N_{2,2}$ are separated horizontally on level 3 to form 4 patches, $I^N_{3,1}$, $I^N_{3,2}$, $I^N_{3,3}$, and $I^N_{3,4}$. At the i-th level, encoders and decoders are labeled as $E_i$ and $D_i$, respectively. In DMPHN, information of features flows from bottom to up. Patches at the lowest level are passed into the encoder $E_3$ to produce feature maps.

$$F_{3,j} = E_3\left(I^N_{3,j}\right), where\ j \in [1,4] \tag{3.1}$$

All these spatially adjacent feature maps are concatenated to get a new feature map representation.

$$R_{3,j} = \left[F_{3,2j-1}, F_{3,2j}\right], where\ j \in [1,2] \tag{3.2}$$

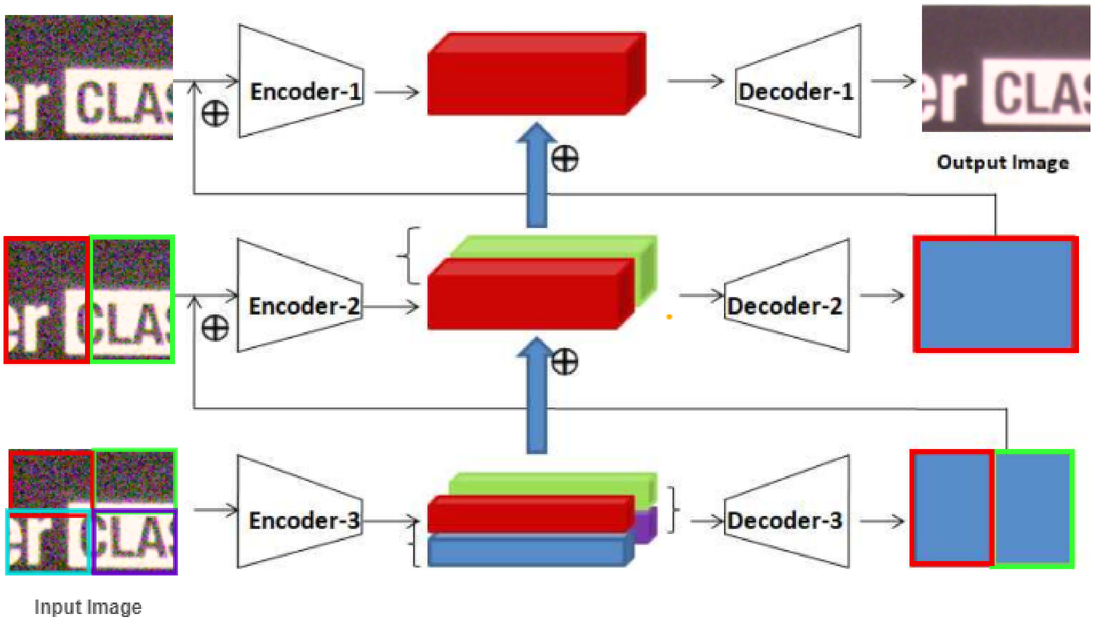


Figure 3.1: Deep Multi-Patch Hierarchical Network (DMPHN) architecture

then these new concatenated features are fed to D3 decoder $D_3$.

$$S_{3,j} = D_3\left(R_{3,j}\right), where\ j \in [1,2] \tag{3.3}$$

this decoder output is added with subsequent level patches and passed to the encoder.

$$F_{2,j} = E_2\left(I^N_{2,j} + S_{3,j}\right), where\ j \in [1,2] \tag{3.4}$$

In the next step, the outputs of the encoder are added to the corresponding decoder inputs from the previous level. After that, the resulting features are concatenated spatially.

$$\begin{aligned} F^*_{2,j} &= F_{2,j} + R_{3,j}, where\ j \in [1,2] \\ R_2 &= \left[F^*_{2,1}, F^*_{2,2}\right] \end{aligned} \tag{3.5}$$

After that,$R_2$ is passed to decoder $\mathrm{D}_2$ to obtain a residual feature map for the second level.

$$S_2 = D_2\left(R_2\right) \tag{3.6}$$

The output of the level-2 decoder is added with the input image and sent via encoder $E_1$. Level-2 adds encoder output $F_1$ to the decoder output.

$$F_1 = E_1\left(I^N + Q_2\right) \tag{3.7}$$

To generate the final clean output image $\hat{I}$, $F_1$ is added to $P_2$ and fed to decoder

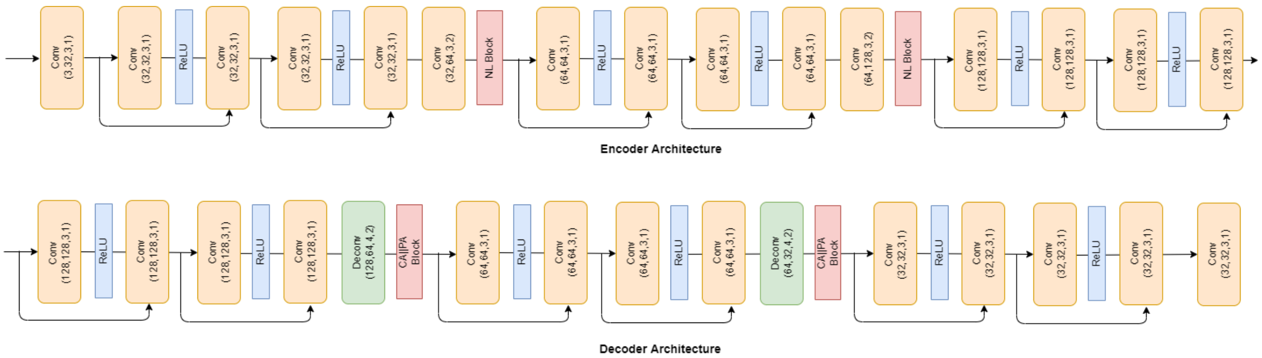


Figure 3.2: DMPHN Encoder and Decoder Architecture

$D_1$.

$$R_1 = F_1 + R_2 \hat{I} = D_1 \left( R_1 \right) \tag{3.8}$$

### 3.3.1 Encoder and Decoder Architecture:

Many computer vision tasks have shown the effectiveness of the symmetrical encoder-decoder architecture [30, 3]. Encoder-decoder networks are symmetrical CNN architectures that gradually transform input images into feature maps with smaller-scale spatial dimensions and a greater number of channels (in the encoder) and then convert it back to the original input shape (in the decoder). Gradient propagation is improved and convergence is accelerated by this design. We employ an encoder to extract features and a decoder to restore images at each level. In encoder-decoders, residual links between the feature maps are extensively used to integrate information of multiple levels. Extra convolution layers are usually added in each level to further improve any network's performance. But [44] has shown that adding extra residual connection within convolutional blocks will improve result quality more than blindly increasing depth by additional convolutions. Hence, the proposed method uses six residual connections in both encoder and decoder modules.

At all levels of DMPHN, we use identical encoder and decoder architecture. Fifteen convolutional layers, two Non-local (NL) block, six residual connections, and six ReLU units are used in the encoder. The decoder architecture is the same as the encoder, except that 2 convolutional layers are replaced with deconvolutional layers to generate a clean output image. Also we replaced NL block with the $CA \parallel PA$ block into decoder part which is a parallel combination Channel Attention (CA) and pixel Attention (PA).

In the architecture, we added NL(non-local) block in the encoders to get rich image features from a entire image instead of considering local patches only. After getting the rich features from the encoder, channel and pixel attention is applied to that features. Because when generating the output, channel and pixel attention determines the relative importance of the set of input features.

### 3.3.2 Non-local Block

**Non-local self-similarity:**

The main idea of classical non-local means [4] is based on the fact that natural images have a lot of redundancy. For example, in Figure 3.3, we can observe that the same color squares have many similarities. So, we can use this non-local self-similarity of the image to infer pixel values.

The algorithm essentially updates each pixel's value by a weighted average of all of the other pixels, with the weight values calculated by considering local neighborhood similarity:

$$I(p) = \frac{1}{C} \sum_{q \in \mathcal{N}} w_{p,q} I(q) \tag{3.9}$$

$\mathcal{N}$ is a neighborhood of the pixel $p$, and $C$ is a normalizing factor. An illustration of the equation (3.1) is shown in the figure 3.4. Equation (3.2) calculates the weights $w_{p,q}$ :

$$w_{p,q} = \exp\left(-\frac{\|\omega(I(p)) - \omega(I(q))\|^2}{h^2}\right) \tag{3.10}$$

where $\omega(p)$ denotes a window centred at pixel $p$.

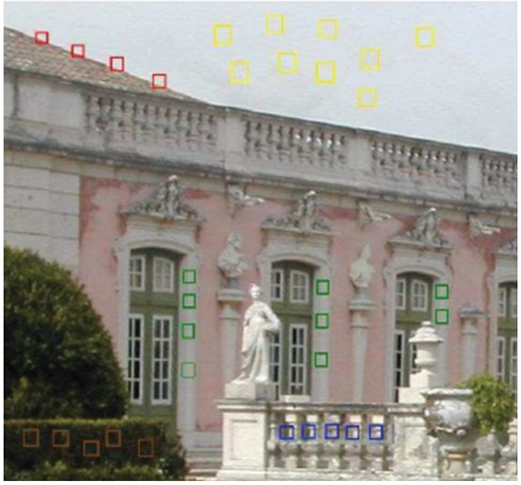


Figure 3.3: Image redundancy

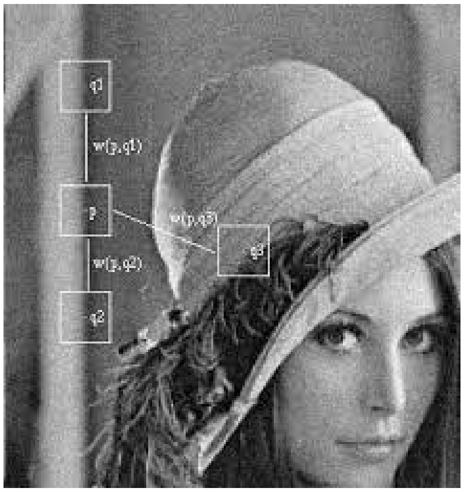


Figure 3.4: NL-means scheme

**Non-local neural network module:**

We used non-local bock presented in [39], which converts above traditional approach into neural network module. It is basically developed for image classification and segmentation tasks to increase receptive field of networks. The architecture of the non-local block is shown in Figure. 3.5.

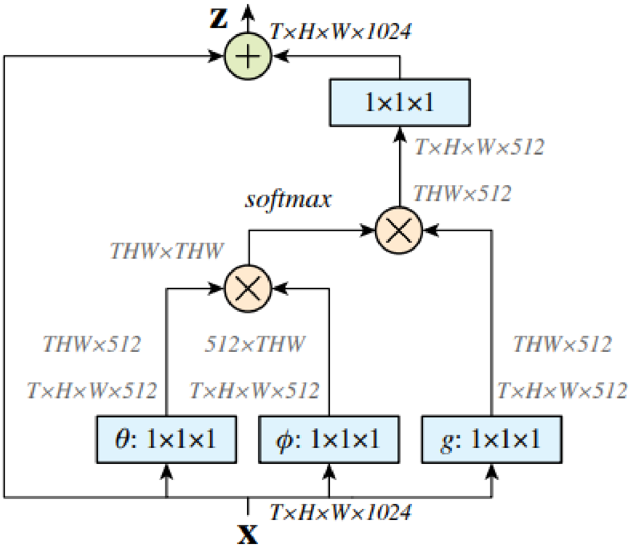


Figure 3.5: Non-local Module

The generic equation for non-local operation in the network is:

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f\left(\mathbf{x}_i, \mathbf{x}_j\right) g\left(\mathbf{x}_j\right) \tag{3.11}$$

where $i$ is the index of an output position and $j$ enumerates all possible positions. $x$ is the input features, and $y$ is the output with a same size as $x$.

$f$ is the function that represents the relationship of a pixel at location $i$ with all the $j$ positions.The Gaussian function is a natural choice for $f$ as it is generally used in non-local mean [4] and bilateral filters [37] for similar scenarios. In this paper we consider: $f\left(x_i, x_j\right) = e^{x_i^T x_j}$ here $x_i^T x_j$ denotes dot product similarity.

$g$ is the unary function used to compute input signal representation at position $j$. It it just a linear embedding, which can be calculated as, $g\left(x_j\right) = W_g x_j$, where $W_g$ is the weight matrix to be learned. The normalization factor C is set as

$$\mathcal{C}(x) = \sum_{j} f\left(x_i, x_j\right) \tag{3.12}$$

so the final equation for the non-local block is $z_i = W_z y_i + x_i$. where $y_i$ is the

Equation-(1) and $x_i$ is a residual link to the output. $W_z$ initially set to 0.

### 3.3.3 Attention Block

The decoder comprises attention blocks. As shown in Figure. 3.6, the attention block combines both channel and pixel attention in parallel (CA || PA).

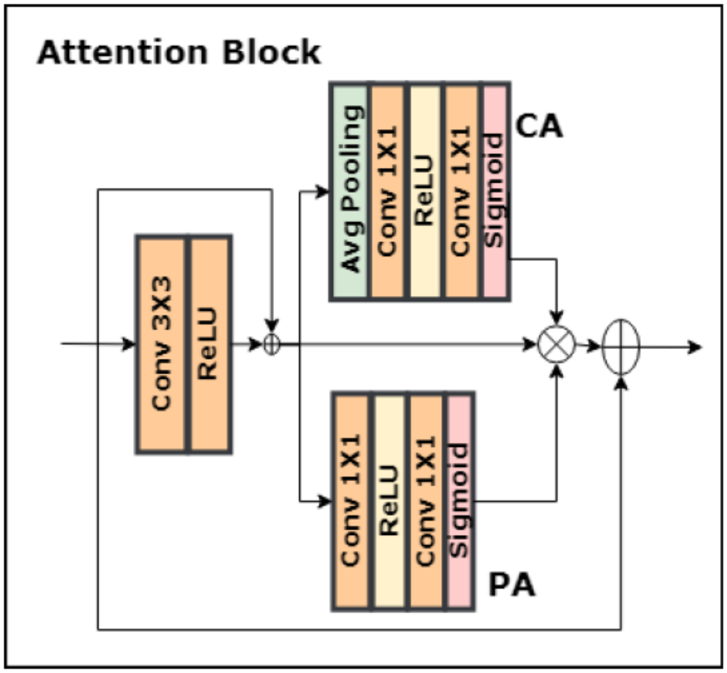


Figure 3.6: Attention Block

The channel attention method defines by

$$A_c = \sigma(MLP(AvgPool(F)) \tag{3.13}$$

Where $F$ represents the input feature. The average pooling procedure on $F$ is used to first aggregates the spatial data. After that, the features are passed into an MLP layer. Finally, the feature is fed to a sigmoid activation function to generate channel attention $A_c$, which multiplies with $F$ as $F_c = A_c \otimes F$. On the contrary, pixel attention $A_p$ is generated via $1 \times 1$ convolution following sigmoid activation $F$. The output of the pixel attention mechanism is $F_p = A_p \otimes F$. The resulting feature map is created by multiplying the output of CA and PA elementwise. The channel attention (CA) reveals crucial information at the channel level, whereas the pixel attention (PA) provides pixel-level attention. In comparison to their serial connections, their parallel connection might better complement each other. The reason for this is that channel attention output will be weighted. As a result, the information at the pixel level is compromised. As a result, pixel attention on the output of channel attention will be ineffective since the PA will be unable to assess the unaffected pixel-level data. Our technique solves this problem by joining CA and PA in a parallel manner.

## Chapter 4

# Implementations and Results

## 4.1 Dataset Description

The SIDD [1] benchmark is used to analyze the denoising performance and quality of our proposed approach in this section. These training images are collected from 160 distinct scene instances, with each instance including two pairs of high-resolution photos, one noisy image, and its associated ground-truth image. So, There are 160 total image pairs. Five different brands' smartphone cameras are used to take these photos with various lighting environments and ISO settings. ISO values varied between 50 to 10,000. The variety of noise levels and lighting conditions in this dataset make it an ideal benchmark for comparing denoising performance. Furthermore, this dataset has a larger size, giving adequate data for training learning-based approaches, particularly convolutional neural network (CNN) based methods.

We used the SIDD small data set for model training, which has over 160 image pairs of clean and noisy images. The validation set is also provided for evaluation, which contains 40 pairs of clean-noisy image patches. During training, a small input size results in faster performance and less memory consumption for GPUs. As a result, we divided each high-resolution image into small patches rather than supplying a whole image to the network. Each image is divided into non-overlapping patches. As a result, we have a training set of 102518 image pairs with a dimension of $120 \times 160$ pixels. The validation data included 1280 noisy image crops, each of with the size $256 \times 256$ pixels. The blocks are made up of 32 blocks from each of 40 images ($40 \times 32 = 1280$). all these blocks represented in a single 4D array of size [40, 32, 256, 256], where the 4 dimension parameters denotes the image index, the block index in particular image, the height, and the width of the block, correspondingly. The blocks are numbered in the same way as the training data.

## 4.2 Training Details

We implemented our model into PyTorch framework. We used Tesla T4 GPU for the training and evaluation. We used the Adam optimizer[16] to train our networks with values of $\beta_1$ and $\beta_2$, 0.9 and 0.99, respectively. The batch size used is 26. The learning rate is 1e-4 at first, then gradually reduced to 5e-5. The model is trained until the point of convergence. We normalized image pixels into [0,1] range and subtract value 0.5 from each pixel.

## 4.3 Loss optimization Function

MSE or L2 loss is a basic loss function used in many denoising methods. It reduces the noise effect but blurs the image details and edges, so we added L1 loss to reduce blurring. Since both L1 and L2 are pixel-wise loss functions and average over all the pixels, it does not consider the human visual perception. Thus we added perceptual loss which considers the image's gray values with texture quality and content quality to generate perceptually better images. Although the good quantitive results, some generated images contain artifacts, so TV loss is used to reduce artifacts while retaining the image sharpness. It ensures spatial continuity and smoothness in the generated image. Hence, we used the combined loss function of the following losses for the model loss optimization.

**Reconstruction loss:** it helps the model to produce denoise images that are near to the ground truth. The weighted sum of the $L_1$ loss and $L_2$ loss is our reconstruction loss. The reconstruction loss is calculated as follows:

$$L_{rec} = \lambda_1 L_1 + \lambda_2 L_2 \tag{4.1}$$

where $L_1 = \|\hat{x} - x\|_1$ and $L_2 = \|\hat{x} - x\|_2$

**Perceptual loss:** it calculated as $L_2$ distance between features of conv4_3 layer of ground truth and predicted images extracted from pretrained VGGnet [32].

$$L_p = \|\phi(\hat{x}) - \phi(x)\|_2 \tag{4.2}$$

**TV loss:** It helps to generate smoother predictions. The loss is calculated as follows:

$$L_{TV} = \|\nabla_X \hat{x}\|_2 + \|\nabla_Y \hat{x}\|_2 \tag{4.3}$$

Where, $\nabla_X$ and $\nabla_Y$ is basically the L2 norm of the image gradient in X and Y directions

The final loss optimization computed as following,

$$L = \gamma_1 L_{rec} + \gamma_2 L_p + \gamma_3 L_{TV} \tag{4.4}$$

the hyperparameters we choose for the experiment are $\gamma_1 = 1, \gamma_2 = 6e - 3$, $\gamma_3 = 2e - 8$. $\lambda_1$ and $\lambda_2$ is chosen to be 0.6 and 0.4 respectively.
Generally, both L1 and L2 losses reduce the pixel-wise dissimilarity. But L2 helps reduce noise while smoothing the image details, and L1 suppresses noise while preserving the edges, so we use a combination of both losses as reconstruction loss with $_1 = 0.6(L1)$ and $\lambda_2 = 0.4(L2)$, instead of giving 0.5 to each.

Only reducing the pixel-wise differences does not give visually good results, so we have given $\gamma_2 = 0.6$ weightage to perceptual loss. This weight is less than the reconstruction loss ($\gamma_1 = 1$) because, while reducing style and content loss the perceptual loss sometimes generates highly pixellated noisy outputs and artifacts too.

TV loss helps reduce artifacts but also smooth the edges if more weightage is given, so we set $\gamma_3 = 0.2$.

For the hyperparameters, we tried two other experiments by two times increasing and two times decreasing all parameter values. The results of these two experiments are shown in the Table 4.1. The table demonstrates that we achieved better results when we set our loss function's hyperparameters to 0.6, 0.4, 0.006, and 2e-8, respectively.

| **Loss function** | ***PSNR*** | ***SSIM*** |
|---|---|---|
| 0.3 * L$_1$ $+ 0.2 * L_2 + 0.003 * perc + 1e - 8 * tv$ | 38.27 | 0.9287 |
| 1.2 * L$_1$ $+ 0.8 * L_2 + 0.012 * perc + 4e - 8 * tv$ | 37.32 | 0.9271 |
| 0.6 * L$_1$ $+ 0.4 * L_2 + 0.006 * perc + 2e - 8 * tv$ | **38.37** | **0.9301** |

Table 4.1: Ablation study: modified weights of the four losses

## 4.4 Evaluation metrics

Many assessment measures [31] are there to determine the quality of the resultant image. Here we will talk about PSNR and SSIM, which are widely used in most denoising methods' evaluations.

### 4.4.1 PSNR

PSNR (peak signal-to-noise ratio) is an objective metric for determining image quality. The higher the PSNR value of the restored image, the closer the denoising image to the clean image, and the error rate is low. PSNR is the inverse of MSE, which can be computed by

$$\mathrm{MSE} = \frac{\sum_{h,w}(G-P)^2}{h \times w}, \tag{4.5}$$

where h and w are the image dimensions, and G and P are the ground truth and predicted images, respectively. if MSE is given then PSNR can be calculated as

$$PSNR = \frac{I^2}{MSE} \tag{4.6}$$

Here, $I$ is the highest possible pixel intensity value. Hence, we use an 8-bit integer representation for each pixel, $I = 255$.

### 4.4.2 SSIM

Structural Similarity (SSIM) is a useful statistic for assessing image similarity. The similarity value of the full image can be determined by evaluating the luminance, contrast, and structural similarity of the image incorporating mean, variance, and covariance. It has a value between 0 and 1. The restored image's higher SSIM value shows that the outcome of denoising is more similar to the ground truth image. It can be computed as follows:

$$SSIM(G,P) = \frac{\left(2\mu_G\mu_P + C_1\right)\left(2\sigma_{GP} + C_2\right)}{\left(\mu_G^2 + \mu_P^2 + C_1\right)\left(\sigma_G^2 + \sigma_P^2 + C_2\right)}, \tag{4.7}$$

where G, P are the identical dimension window of the ground truth image and the predicted image respectively. $\mu_G$ and $\mu_P$ are the mean values of G and P, correspondingly. $\sigma_G$ and $\sigma_P$ are the variances of G and P, respectively . $\sigma_{GP}$ is the covariance of G and P . $C_1$ and $C_2$ are both constants.

## 4.5 Geometric Self-ensemble

To enhance the overall model performance further, we use the self-ensemble technique, as described in [20]. During the testing phase, the input image $I^N$ is flipped and rotated to produce seven augmented input images $I_{n,i}^N = T_i\left(I_n^N\right)$ for each sample, in which $T_i$ indicates the eight geometric variations including self image. Using the model, we produce equivalent denoised images $\left\{ I_{n,1}^{D}, I_{n,2}^{D}, \cdots, I_{n,8}^{D} \right\}$ from the given augmented noisy images. The initial basic geometry $\tilde{I}_{n,i}^{D} = T_i^{-1}\left( I_{n,i}^{D} \right)$ is then obtained by applying an inverse transformaion to the resulting images. At last, we combine the transformed output images by taking average, to obtain the self-ensemble result shown below:

$$I_n^D = \frac{1}{8} \sum_{i=1}^{8} \tilde{I}_{n,i}^{D} \tag{4.8}$$

The self-ensemble approach has a benefit over the other ensemble methods since it does not need the additional model training. It is highly beneficial when the model's size or preparation time is important. Even though the self-ensemble technique maintains the overall number of parameters quite equivalent, we find that it provides nearly the same performance improvement as the traditional model ensemble method, which needs separate model training.

## 4.6 Results

The 256 x 256 pixel validation patches are fed into DMPHN for model evaluation. Table 4.1 shows the PSNR, SSIM, and number of network parameters for a detailed comparison with competitive state-of-the-art denoising methods. As shown in Table 4.1, our proposed method gives better PSNR results and significantly moderate results for SSIM compared to other methods, demonstrating the effectiveness of real-world noise removal through the use of localize information in the model. It is worth noting that our model is simpler than competing methods, as we do not use any dense modules. In the table bold values represent the best result, and underlined values represent the 2nd best result among all the values in the particular column.

Figures 4.1 and 4.2 shows denoised images from the SIDD dataset. Figure 4.1 shows the denoising performance of various models on an image with high brightness and excessive noise. For clarity, we zoom in on a small section of the image. In Figure. 4.2, we have chosen an image with very low light and heavy noise since many methods do not perform well in this scenario. It can be observed that our proposed method removed most of the noise from the image and also recovered the edges better than other methods.

Moreover to our model's higher PSNR and SSIM, DMPHN is, to the best of our knowledge, the best deep denoising method that can function in real-time also. For instance, It requires 30ms for processing a 720 x 1280 image, implying that it provides real-time 720p image denoising at 30fps. However, because of the runtime overhead costs associated with I/O operations, the real-time denoising system requires quick transfers between a video grabber and GPU, greater GPU memory capacity and an SSD disk, and so on. Our faster runtime is due to the following factors: 1) shorter encoder-decoder with small scale convolutional filters; 2) elimination of unwanted linkages, such as skip or recurrent links; and 3) lower number of upsampling/downsampling between convolutional features of various levels.

Compared with other approaches, we use shallower CNN encoders-decoders with smaller model sizes (only 21.7 MB) and parameters than all other methods. This is due to our simplified CNN architecture.

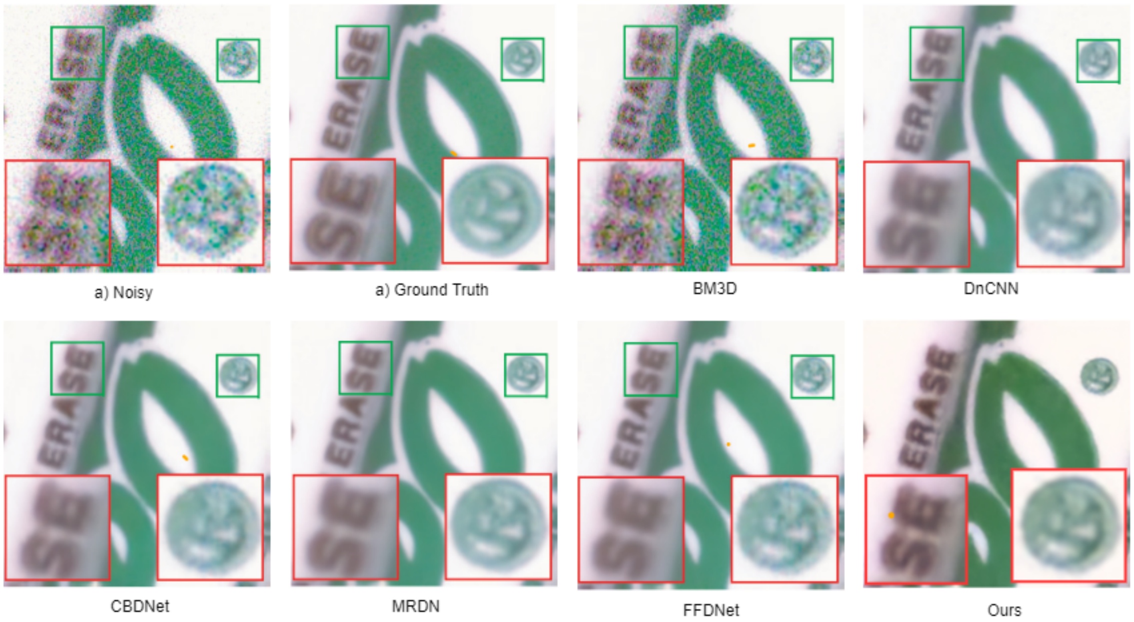
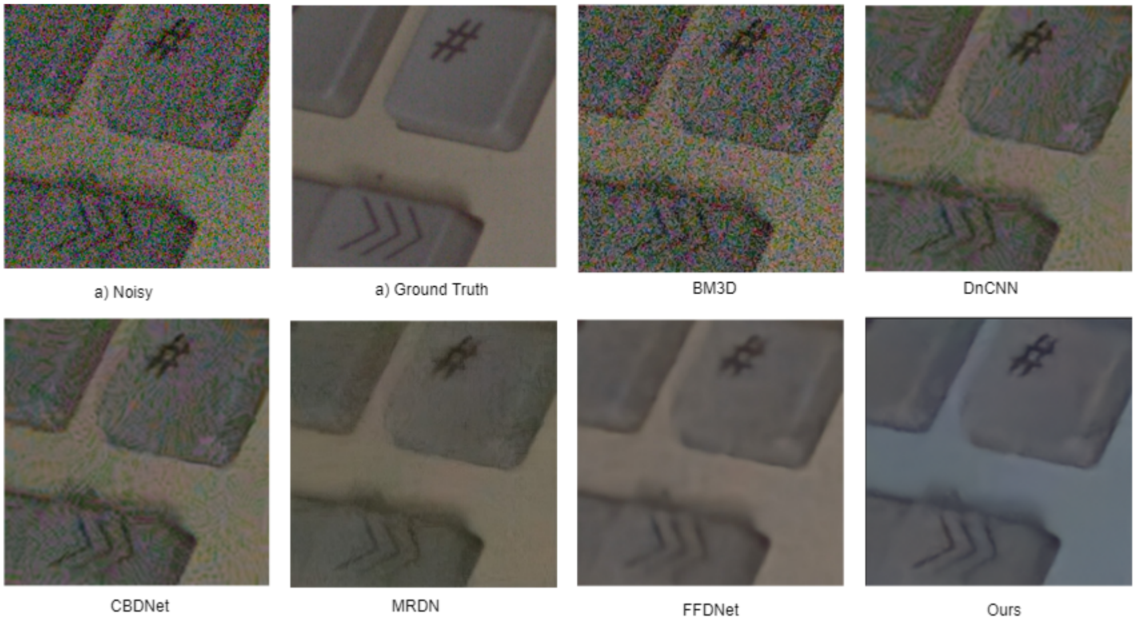


Figure 4.1: Qualitative results on SIDD Validation Dataset (Ex.1)



a) Noisy

a) Ground Truth

BM3D

DnCNN

CBDNet

MRDN

FFDNet

Ours

Figure 4.2: Qualitative results on SIDD Validation Dataset (Ex.2)

| **Methods** | **PSNR** | **SSIM** | **# of parameters** |
|---|---|---|---|
| BM3D [5] | 25.65 | 0.685 | - |
| NLM [4] | 26.75 | 0.699 | - |
| DnCNN [46] | 30.71 | 0.695 | 558K |
| CBDNet [11] | 33.28 | 0.868 | 4347K |
| MRDN [3] | 36.42 | 0.875 | 485K |
| MCUNet [3] | 36.54 | 0.878 | 1499K |
| FFDNet [47] | 38.28 | **0.948** | 685K |
| Ours | **38.37** | 0.931 | **449K** |

Table 4.2: Quantitative results on SIDD Validation Dataset

We test our technique on another real-world benchmark, the Nam dataset [24], to evaluate its robustness in general real-world denoising tasks. Also, the results are compared with many state-of-the-art methods. These comparative results are shown in table 4.3.

| Metrics | BM3D | NLM | DnCNN | CBDNet | FFDNet | Ours |
|---|---|---|---|---|---|---|
| PSNR | 35.36 | 35.33 | 35.68 | 39.20 | 38.57 | 38.49 |
| SSIM | 0.8708 | 0.8812 | 0.8811 | 0.9676 | 0.9570 | 0.9759 |

Table 4.3: Results on Nam Dataset [24]

More qualitative examples for different scenarios are shown in Figure 4.3.

Here, images 1 and 2 are captured with very low light, whereas images 3 and 4 are captured in excessive light, and the last image contains very high noise. The result demonstrates that the noise is almost removed, and edges are preserved well for each case.

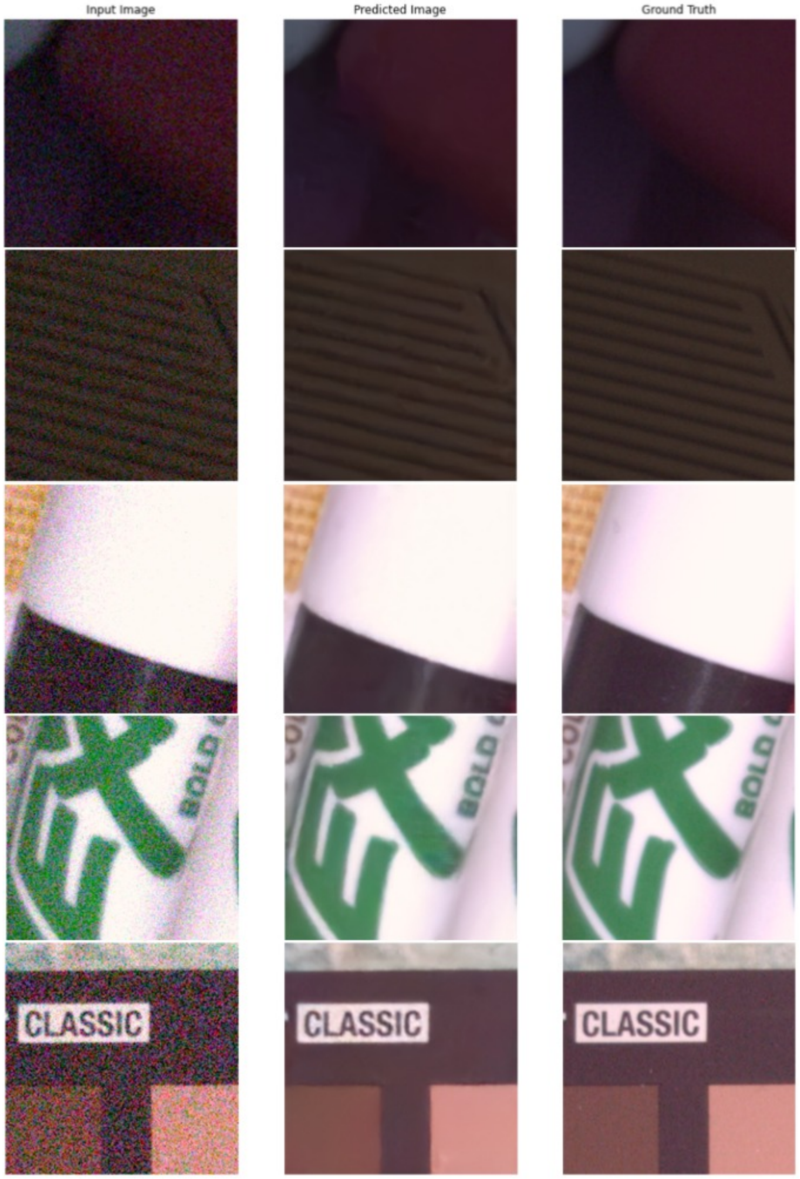


Figure 4.3: Qualitative results on different scenarios (SIDD Dataset)

## 4.7 Additional Experiments

### 4.7.1 SE Block

In 2018 J. Hu et al. [13] introduced SE (Squeeze-and-Excitation) block. The objctive of block is to to increase a network's representation capability by explicitly modeling the inter-dependencies between its convolutional features' channels. To accomplish this, they presented a technique that enables the network to execute feature recalibration, allowing it to learn using global information to selectively enhance important features while suppressing less relevant ones. We incorporated this block into network as it captures channel-wise feature dependency very well.

### 4.7.2 NLSA Module

The advantages of sparse representation and non-local operation are combined in Non-Local Sparse Attention (NLSA) [23]. NLSA globally determines the most informative regions without considering unrelated areas, resulting in an efficient and reliable global modelling operation. Their evaluations suggested that NLSA seems to be a better operation than normal non-local attention when inserted into deep networks.

| **Experiments** | **PSNR** | **SSIM** |
|---|---|---|
| SE block [13] | 38.21 | 0.9293 |
| NLSA module [23] | 37.65 | 0.9132 |

Table 4.4: Experimental results on SIDD Validation Dataset

# CHAPTER 5
# Conclusion & Future Scope

## 5.1 Conclusion

We propose a deep multi-patch hierarchical network, which performs denoising of the image by aggregating features generated from finer to coarser levels, and we observe it is well suitable for the denoising task. Also, we demonstrate the significance of the non-local module, which uses the non-local self-similarity of natural images. Major improvement in results is achieved by simply adding non-local blocks into the network. We also suggest channel and pixel attention to enhance the network's capacity to capture inter-dependencies between channels and pixels. In terms of qualitative and quantitative evaluation, our proposed architecture generates denoised results that are equivalent to the state-of-the-art methods.

## 5.2 Future Work

Nowadays, vision transformers [8] are defeating state-of-the-art performances of many image processing tasks, so we can try using transformer blocks in the network instead of convolutional blocks. Also, we can attempt multi-scaling coarse-to-fine approaches to find global and local information at a different scale. The model's capability can be checked over specific noise such as thermal, dark current, or shot noise.

# References

[1] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018.

[2] J. Anaya and A. Barbu. Renoir–a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018.

[3] L. Bao, Z. Yang, S. Wang, D. Bai, and J. Lee. Real image denoising based on multi-scale residual dense block and cascaded u-net with block-connection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 448–449, 2020.

[4] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[6] S. D. Das and S. Dutta. Fast deep multi-patch hierarchical network for non-homogeneous image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 482–483, 2020.

[7] S. D. Das and S. Dutta. Fast deep multi-patch hierarchical network for non-homogeneous image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 482–483, 2020.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[10] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

[11] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[15] W. Jifara, F. Jiang, S. Rho, M. Cheng, and S. Liu. Medical image denoising using convolutional neural network: a residual learning approach. *The Journal of Supercomputing*, 75(2):704–718, 2019.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[18] S. Lefkimmiatis. Universal denoising networks: a novel cnn architecture for image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3204–3213, 2018.

[19] A. Levin, B. Nadler, F. Durand, and W. T. Freeman. Patch complexity, finite pixel correlations and optimal denoising. In *European Conference on Computer Vision*, pages 73–86. Springer, 2012.

[20] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. *CoRR*, abs/1707.02921, 2017.

[21] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018.

[22] M. Mastriani, A. Giraldez, et al. Microarrays denoising via smoothing of coefficients in wavelet domain. *arXiv preprint arXiv:1807.11571*, 2018.

[23] Y. Mei, Y. Fan, and Y. Zhou. Image super-resolution with non-local sparse attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2021.

[24] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1683–1691, 2016.

[25] M. Noroozi, P. Chandramouli, and P. Favaro. Motion deblurring in the wild. In *German conference on pattern recognition*, pages 65–77. Springer, 2017.

[26] T. Plotz and S. Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[28] S. Ren, J. Sun, K. He, and X. Zhang. Deep residual learning for image recognition. In *CVPR*, volume 2, page 4, 2016.

[29] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[31] U. Sara, M. Akter, and M. S. Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[33] V. Soni, A. K. Bhandari, A. Kumar, and G. K. Singh. Improved sub-band adaptive thresholding function for denoising of satellite image based on evolutionary algorithms. *IET Signal Processing*, 7(8):720–730, 2013.

[34] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik. Compression artifacts removal using convolutional neural networks. *arXiv preprint arXiv:1605.00366*, 2016.

[35] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[36] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin. Deep learning on image denoising: An overview. *Neural Networks*, 2020.

[37] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, 1998.

[38] G. Vaksman, M. Elad, and P. Milanfar. Lidia: Lightweight learned image denoising with instance adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 524–525, 2020.

[39] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[40] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018.

[41] C. Yu and X. Chen. Remote sensing image denoising application by generalized morphological component analysis. *International journal of applied earth observation and geoinformation*, 33:83–97, 2014.

[42] S. Yu, B. Park, and J. Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[43] Y. Yu, H. Liu, M. Fu, J. Chen, X. Wang, and K. Wang. A two-branch neural network for non-homogeneous dehazing via ensemble learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 193–202, 2021.

[44] H. Zhang, Y. Dai, H. Li, and P. Koniusz. Deep stacked hierarchical multipatch network for image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5978–5986, 2019.

[45] H. Zhang, Y. Dai, H. Li, and P. Koniusz. Deep stacked hierarchical multipatch network for image deblurring. *CoRR*, abs/1904.03468, 2019.

[46] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

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

[48] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, and S. Howard. A poisson-gaussian denoising dataset with real fluorescence microscopy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11710–11718, 2019.

[49] W. Zuo, K. Zhang, and L. Zhang. Convolutional neural networks for image denoising and restoration. In *Denoising of Photographic Images and Video*, pages 93–123. Springer, 2018.