UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Telecommunications Engineering

A Study on Image Denoising Using Deep Learning

End of Degree Project

Bachelor's Degree in Telecommunication Technologies and Services Engineering

AUTHOR: Lin, Guanyi

Tutor: Vergara Domínguez, Luís

External cotutor: SALAZAR AFANADOR, ADDISSON

ACADEMIC YEAR: 2023/2024

Undergraduate Project Report
2023/24

A Study on Image Denoising Using Deep Learning

Name:	Guanyi Lin
School:	International School
Class:	2020215102
QMUL Student No.:	200977962
BUPT Student No.:	2020213069
Programme:	Telecommunications Engineering with Management

Date: 24-04-2024

Table of Contents

Abstract

In the process of image transmission and reception, digital images inevitably encounter noise interference, which adversely affects image quality. The purpose of image denoising is to effectively remove noise from images while retaining the original features as much as possible. Extensive research indicates that deep learning methods achieve superior outcomes in image denoising. This project conducts a thorough investigation and categorized summary of various denoising algorithms based on deep learning (DL), completing a literature review. On this basis, the paper successfully proposes and trains an improved model of the DnCNN-based image denoising network on the MATLAB platform. It conducts a series of comparative experiments on multiple types of noise including Gaussian, Exponential, Rayleigh, Salt-and-Pepper, Speckle, Poisson and K-distribution noise, as well as different noise amplitudes, comparing the denoising effects of traditional image denoising filters with different convolutional neural networks (CNN) models. The effectiveness of denoising is validated through evaluation metrics such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM).

My study demonstrates the robustness and versatility of CNN in image denoising, and the proposed CNN significantly enhances denoising performance, with broad application prospects in the field of digital image processing, notably in improving the accuracy of image classification.

Keywords

Image denoising, deep learning methods, convolutional neural networks (CNN), literature review, MATLAB, digital image processing, types of noise, noise amplitudes, DnCNN, improved model, comparative experiments, image quality metrics, application prospects..

摘要

This is the Chinese translation of the Abstract.

数字图像在传输和接收过程中总会受到噪声干扰，对图像质量产生影响。图像去噪的目的就是在尽可能地保留原始特征的前提下有效去除图像噪声。大量研究表明，深度学习方法对于图像去噪有着更好的效果，本项目对基于深度学习的各类去噪算法进行了充分的调查和分门别类的总结，并完成文献综述。在此基础上，基于 MATLAB 平台，本文提出并训练了基于 DnCNN 的图像去噪网络改进模型，并且对高斯噪声、指数噪声、瑞利噪声、椒盐噪声、斑点噪声和 K 分布噪声等多种噪声类型及不同噪声幅度进行一系列对照实验，即分别使用传统的图像去噪滤波器和不同卷积神经网络(CNN)模型进行了去噪效果比较分析。最后通过均方误差（MSE）、峰值信噪比（PSNR）和结构相似性指数（SSIM）等评价指标验证去噪效果。

我的研究表明 CNN 在图像去噪方面具有鲁棒性和多功能性等特征，所提出的 CNN 有效提高了去噪性能，拥有广泛应用前景，例如数字图像处理领域中提高图像分类准确性等。

关键词

This is the Chinese translation of the Keywords.

图像去噪、深度学习方法、卷积神经网络、文献综述、MATLAB、数字图像处理、噪声类型、噪声幅度、DnCNN、改进模型、对比实验、图像质量指标、应用前景

Chapter 1: Introduction

In the field of digital image processing, the presence of image noise is a pervasive challenge that compromises the clarity and quality of images, directly blurring the details of images and impacting their analysis and subsequent applications, such as image classification and recognition. As a result, the main purpose of image denoising is to effectively remove noise while minimizing the loss of original features as much as possible after image capture or transmission.

To better tackle this problem, my project focuses on applying CNN and deep learning techniques, which are highly promising in removing noise from images. Unlike old methods such as noise reduction filters, CNN is better at telling the difference between noise, edges, and textures, despite their similar high-frequency features. This ability makes CNN adaptable to various noise removal scenarios, from ones needing guidance to those that are more self-reliant, and from common image editing to specific tasks.

In addition, an important part of this project is using MATLAB for its setup. MATLAB is excellent for many image and data handling tasks, with advanced features and better speed than Python in many cases. It has many unique tools, avoiding the common data type issues found with Python. MATLAB's strong community support adds to its value, offering help and a chance to share knowledge. Importantly, MATLAB supports different frameworks like CNN and UNet, making it a great choice for this project.

Furthermore, CNN significantly improves image denoising, outperforming traditional methods by effectively removing various types of noise. Their integration with MATLAB enhances this process, leveraging CNN's capabilities within MATLAB's robust environment. This project aims to develop a comprehensive denoising solution, contributing to image processing advancements, particularly in classification. Details on the methodology and results are discussed in subsequent chapters.

The report is structured to provide a comprehensive overview of the project, beginning with a detailed literature review of the background in image denoising methods, especially CNNs in

Chapter 2, It outlines various noise types and traditional noise reduction techniques, highlighting CNNs' wide use across different applications. The discussion includes supervised, self-supervised, and unsupervised denoising methods, introducing cutting-edge approaches These sections reflect the depth of research and the novel contributions made towards advancing the state of the art in image denoising .Through this discussion, the background review of image denoising methods sets the stage for the subsequent development and implementation of an improved CNN-based denoising framework.

Following the background literature review, my design and implementation of the CNN for denoising are thoroughly documented in Chapter 3. This section delves into the architecture of the network, detailing the modifications and enhancements introduced to the basic DnCNN model to tailor it for handling a diverse array of noise types and amplitudes. The chapter provides insights into the dataset preparation, noise addition techniques, and the training process, culminating in a depiction of the novel network architecture designed for this project.

Chapter 4 presents the results and discussions derived from the application of the developed CNN model. It evaluates the denoising performance using a suite of metrics, including MSE, PSNR, and SSIM, to scientifically measure the effectiveness of the implemented techniques. This section offers a critical analysis of the denoising outcomes, comparing them with traditional denoising filters and discussing the implications of these results for image classification accuracy, which demonstrates the better denoising effect of my trained denoising CNN compared to other traditional denoising filters and CNN like DnCNN.

Chapter 5 concludes the project's journey, highlighting its advancements in CNN-based image denoising and its impact on image classification accuracy. It reflects on the technical skills honed, deeper insights into deep learning for denoising, and the importance of ethical research practices. Future directions include leveraging PyTorch and cloud computing for enhanced computational power, exploring unsupervised learning methods for practical training scenarios, and optimizing models for broader applications in image processing.

Chapter 2: Background Literature Review of Image Denoising Methods

NOISE in images often occurs as isolated pixels or pixel blocks that have a significant visual impact, disrupting the actual information content of the image and making it unclear, which makes it difficult to analyze and negatively impacting subsequent tasks such as image detection, classification, segmentation, tracking and more. So, current image processing often challenges prominently include the effective removal of noise - a disruption that obscures critical information content. This review concentrates on recent advancements in image denoising, emphasizing the application of Deep Learning (DL) methodologies. I explore various noise types (e.g., white Gaussian noise, Poisson noise) and filters (e.g., linear, non-linear, adaptive) used for noise mitigation. Significantly, Convolutional Neural Networks (CNNs) are highlighted for their adaptivity, efficiency, and capability to handle complex noise patterns, outperforming traditional filters. It categorizes CNN-based denoising strategies for general and specific imagery, and supervised, self-supervised and unsupervised, detailing their applications, advantages, and the challenges they address. Through this exploration, I underscore the critical role and continual evolution of DL and CNNs in advancing image denoising techniques, which helps the code implementation.

2.1 Image Noise & Noise Reduction Filters

Noise can be introduced at different image processing stages when inputs diverge from expectations. One of the great problems in image denoising is to distinguish the noise, edge and texture, which all have high-frequency components. To further discuss, people usually categorize the noise as additive white Gaussian noise (AWGN), impulse noise, quantization noise, Poisson noise, salt-and-pepper noise and speckle noise. The examples of noisy images can be seen in Appendix. Among them, the AWGN often occurs in analog circuitry (E.g. information channels) while the rest of them occur due to faulty manufacturing, bit error, and inadequate photon count[2].

To mitigate various types of noise, a range of noise reduction filters have been developed, each with its own method of smoothing out disturbances while preserving important image details. These filters fall into six principal categories: linear, non-linear, adaptive, wavelet-based, partial differential equation (PDE), and total variation filters.

Linear filters, which perform noise reduction by correlating output pixels with their neighboring inputs using matrix operations, often blur edges, resulting in a loss of sharpness. On the other hand, non-linear filters like the median filter maintain edge integrity while dampening noise, making them a go-to choice for many applications.

For real-time processing, adaptive filters stand out. They employ statistical methods to dynamically adjust to changing noise patterns, and Wavelet-based filters offer a different approach, transforming images into the wavelet domain to target and reduce additive noise.

Despite the effectiveness of these filters in various scenarios, they come with challenges. They may not perform optimally during testing phases, often require manual tuning of parameters, and their reliance on specific denoising models can limit their applicability across different types of noise environments. These limitations highlight the importance of selecting the appropriate filter based on the noise characteristics and the application's requirements[3].

Filter Type	Denoising Type	Image Quality Comparison	Formula and Computational Complexity	Additional Notes
Gaussian Filter	Normal distribution noise (AWGN)	More effective with increasing standard deviation, but causes more blurring	Convolution operation, implemented in time or frequency domain, slower computation	Separable
Mean Filter	Various types of noise	Becomes blurrier with larger kernel sizes	Simple formula, lower computational complexity	Linear filtering method
Median Filter	Especially effective for salt-and-pepper noise	Better edge preservation while denoising	Depends on sorting algorithms, complexity	Non-linear filtering method

Filter Type	Denoising Type	Image Quality Comparison	Formula and Computational Complexity	Additional Notes
			varies with implementation	
Bilateral Filter	Various types of noise except salt-and-pepper noise	Best edge preservation while denoising	Includes spatial and range matrices, more complex computation	Preserves high-frequency information like edges, suitable for complex background images

TABLE 1: Some common filters and their characteristics

2.2 Advantages of using CNN in image denoising

To overcome the limitations of traditional noise reduction filters, specifically in image denoising tasks, Convolutional Neural Networks (CNN) is widely used due to the flexibility and advancements:

2.2.1 Adaptivity and Learning: Unlike traditional filters, where parameters are fixed and cannot be adjusted during the filtering process, CNN can learn and optimize their parameters through network training. This adaptability allows CNN to perform better in various and changing noise conditions.

2.2.2 Handling Complex Noise Patterns: CNNs are better suited for handling complex noise patterns, which traditional filters might not handle all kinds of them effectively. This is due to the learning capability of CNNs, which can adapt to various noise distributions in the training data.

2.2.3 Preservation of Image Features: CNNs, especially those designed for image denoising, are often more effective in preserving important image features, such as edges and textures, while removing noise. This is a critical advantage over some traditional filters that might blur or distort these features.

2.2.4 Efficiency in Processing Large Datasets: With the advancement in computational power and the availability of large image datasets, CNNs can be trained more effectively and can process large amounts of data more efficiently than traditional methods.

2.2.5 Versatility and Integration: CNNs can be integrated into broader image processing and computer vision pipelines, offering versatility in applications ranging from basic image enhancement to complex tasks like object detection and scene analysis[1].

The unique and critical advantage of using CNNs for image denoising lies in their ability to learn from data, adapt to various noise patterns, and preserve important image features while efficiently processing large volumes of data. These aspects make CNNs a forward-looking technology in the field of image processing, with potential for continued improvements and innovations.

2.3 Classification of CNNs by application fields

Overall, CNN architectures have been enhanced with components such as convolution layers, batch-normalization, ReLU activation functions, and residual learning to improve denoising performance.

Early CNN development faced challenges such as the vanishing gradient problem and hardware limitations, which were overcome by significant advancements like AlexNet in 2012[4], followed by other architectures such as VGG, ResNet[5] and UNet, GoogleNet.

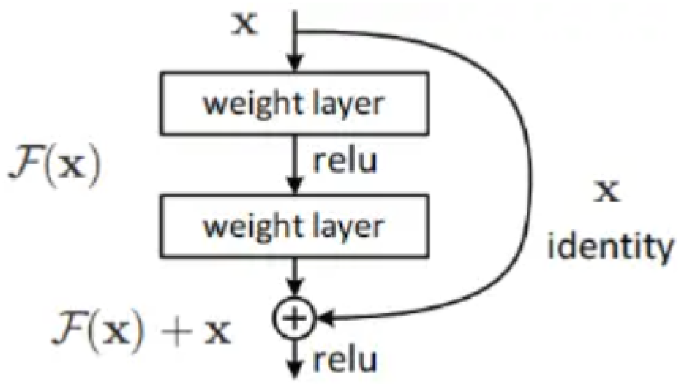


FIGURE 1: A building block of ResNet [5]

Nowadays, more image denoising specified CNNs are introduced to use. For example, Zhang et al. ‘s DnCNN architecture is highlighted for its application in image denoising, super-resolution, and JPEG image deblocking, which can be directly use in MATLAB[6]. This denoising convolutional neural network was trained using manually added AWGN to create noisy-clean image pairs. DnCNN's use of residual learning not only enhanced denoising performance but also significantly reduced computational demands.

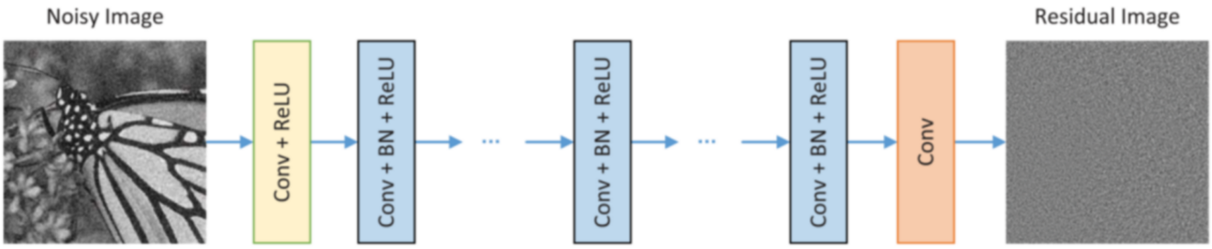


FIGURE 2: The architecture of DnCNN [6]

These CNNs have been shown to effectively address drawbacks such as suboptimal test phase performance, manual parameter settings, and model specificity in image denoising comparing with the traditional filters. Also, the capabilities of CNNs extend beyond image denoising to applications in image recognition, robotics, self-driving cars, facial expression recognition, natural language processing, and handwriting digital recognition[1].

Building upon this versatile foundation, CNNs for image denoising can be systematically categorized into two distinct strategies by application fields:

1. General images (for general proposes), which are utilized for a broad spectrum of purposes. Here, CNN architectures are employed to clean noise from images that are intended for general use.
2. Specific images (for detailed and specialized use) , which are crafted for detailed and specialized applications. In this case, CNNs are tailored to filter noise from images where intricate details are crucial.

These two strategies demonstrate the adaptable nature of CNNs, as they are engineered to handle both the overarching needs of general imagery and the precise requirements of specialized images.

The classification of CNN denoising methods into categories based on image type serves to acquaint the audience with the most current CNN architectures tailored for various image classifications. For a visual representation of these methodologies, refer to the block diagram presented in Figure 1. This delineation sets the stage for a detailed exploration of the distinct approaches applied to both general and specific image denoising, each adapted to its respective image category's requirements.

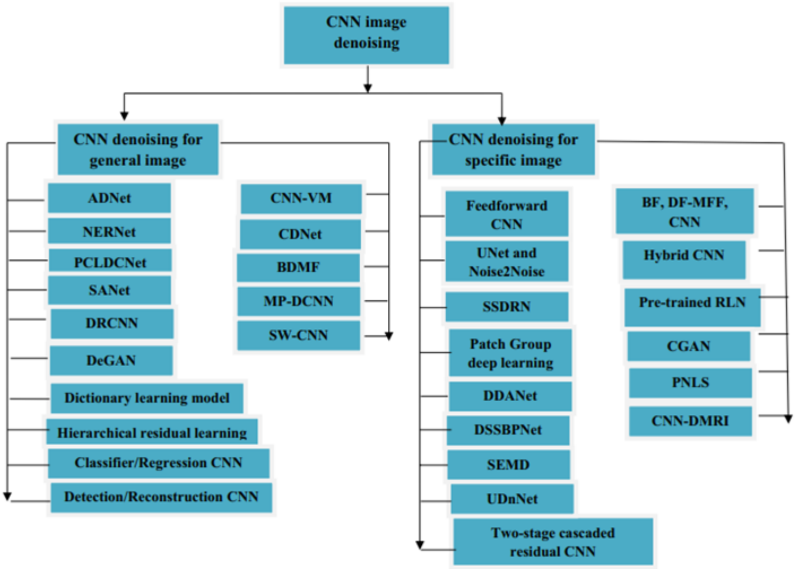


FIGURE 3: CNN image denoising scheme [1]

2.3.1 CNNs for general image denoising

There are many CNNs to deal with the general images denoising.

Attention-guided denoising CNN(ADNet) is structured into four distinct blocks over 17 layers: the Sparse Block (SB) for enhanced efficiency and reduced depth with 12 layers, the Feature Enhancement Block (FEB) to amplify features through its 4 layers, the single-layer Attention Block (AB) to focus on unknown noise, and the Reconstruction Block (RB) for final image output. Sparsity in SB and attention in AB refine the denoising, while mean square error guides the training[7].

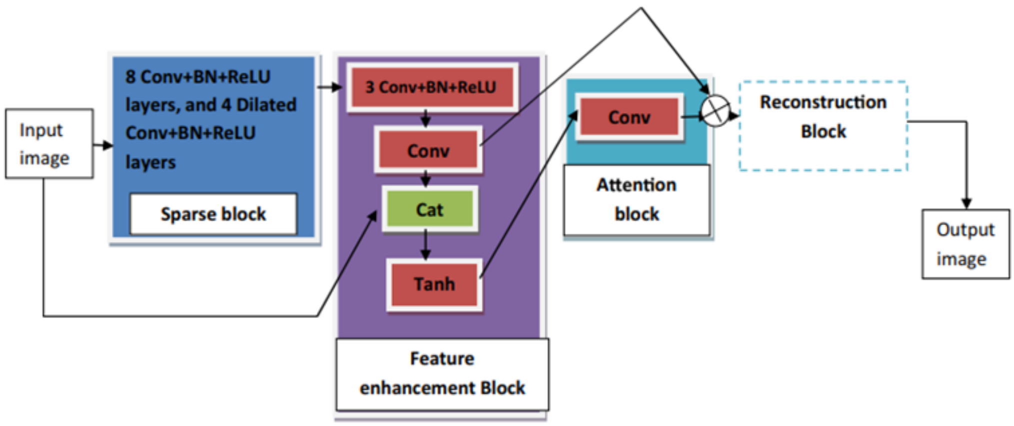


FIGURE 4: Attention-guided denoising CNN [7]

Some CNNs can only get good results of synthetic noise instead of realistic noise. To solve this problem, Noise Estimation and Removal Network (NERNet) was introduced, which consists of two core modules for addressing real noise in images: the noise estimation module utilizes symmetric dilated blocks and pyramid feature fusion to gauge noise levels, while the noise removal module employs this estimation to eliminate noise, merging global and local insights to retain image details and textures. [1]

CNN's proficiency in learning noise patterns and image patches necessitates extensive training data, leading to the creation of the patch complexity local divide and deep conquer network (PCLDCNet). This network segments the learning process into local subtasks, focusing on individual clean image patches, enabling efficient training within their specific local domains[1].

Another problem of CNN is that the deeper the layer, the higher the error rate (Network Degradation), ResNet is designed to solve it. To make better use of ResNet, Patch Complexity Local Divide and Deep Conquer Network((PCLDCNet) is introduced. The network is partitioned into local subtasks based on clean image patch and conquer block and is trained in its local space. Then each noisy patch weighting mixture is amalgamated with the local subtask, and finally fuse with noise map to estimate[1].

To further improve, MP-DCNN, an adaptive residual CNN that operates end-to-end. It utilizes leaky ReLU to extract noise and reconstructs image features. Incorporating SegNet, it retrieves edge details from an initial denoised image. The model employs both MSE and a perceptual loss function to produce the final denoised output[8].

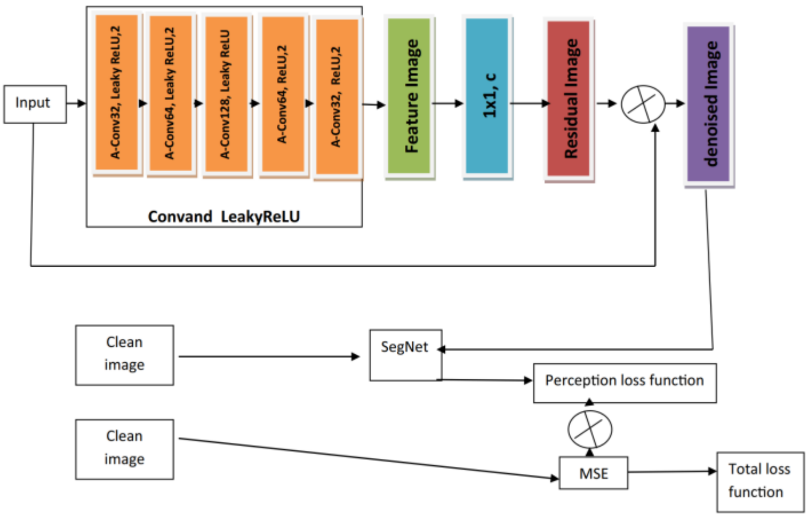


FIGURE 5: MP-DCNN [8]

Furthermore, removing mixed noise from images using CNN is more challenging than addressing single noise types. So DeGAN, a denoising-based generative adversarial network, is introduced to tackle mixed noise. GANs are already established in deep learning, and DeGAN integrates a generator, discriminator, and feature extractor[9].

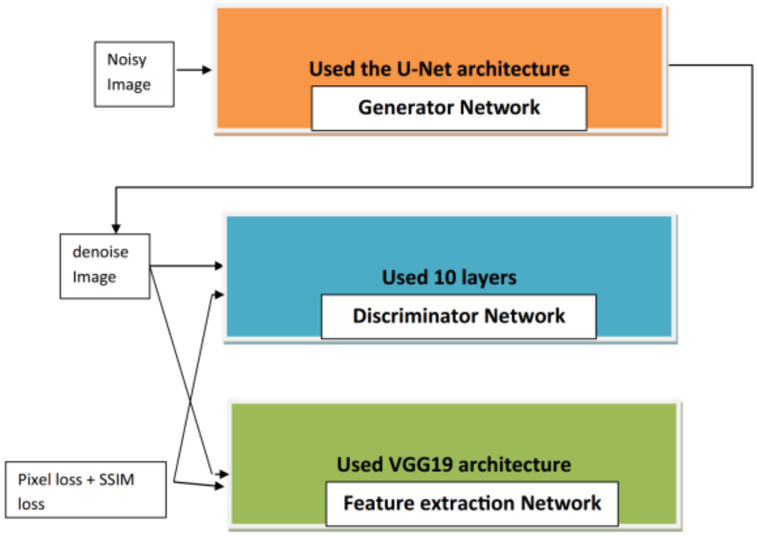


FIGURE 6: DeGAN [9]

To combine classification model and regression model together, a classifier/regression CNN framework for image denoising is introduced. In this system, the classifier network, equipped with convolution, BN, ReLU, softmax, and a skip connection, detects impulse noise. This restoration is guided by the classifier's predictions, aiming to reconstruct clean images[10].

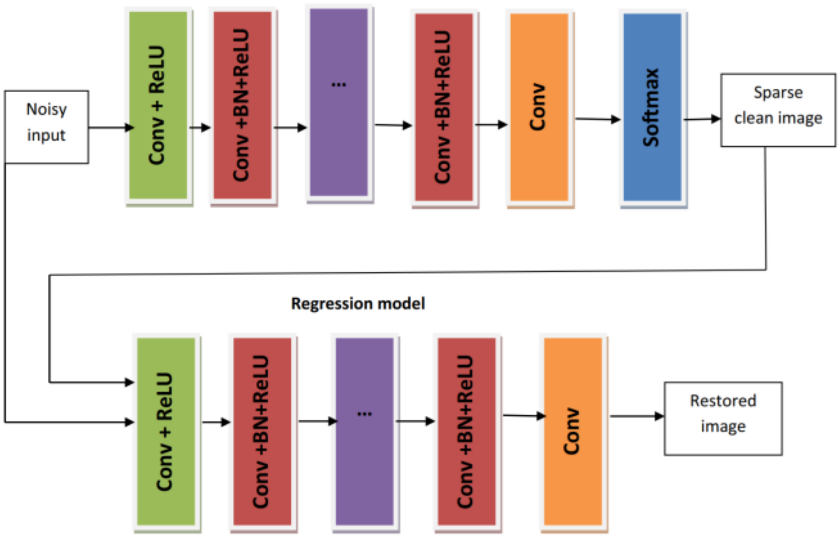


FIGURE 7: Classifer/regression CNN [10]

CDNet is a complex-valued CNN for image denoising. The process begins with feeding the input image into 24 Sequentially Connected Convolutional Units (SCCU), each comprising a complex-valued (CV) convolutional layer, CV ReLU, and CV Batch Normalization (BN). CDNet is structured into five key blocks: CV Convolution, CV ReLU, CV BN, CV Residual Block (RB), and the merging layer[11].

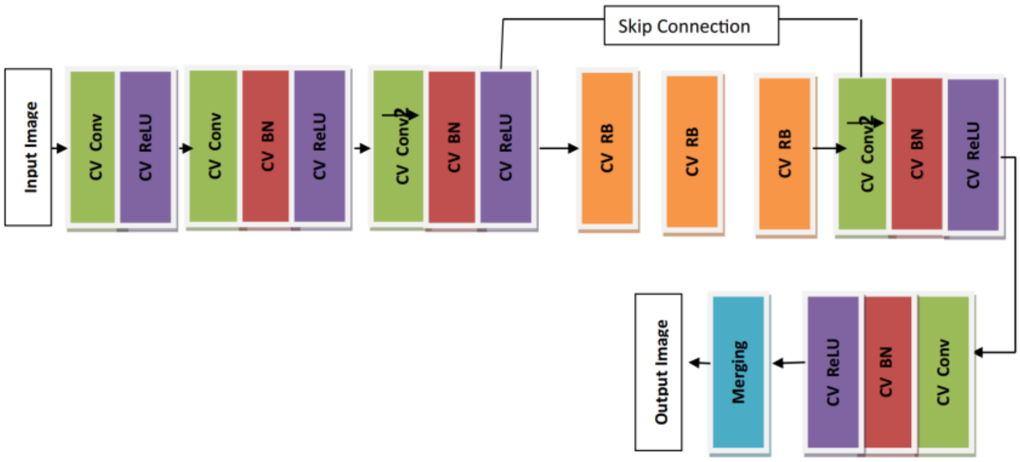


FIGURE 8: Complex value CNN [11]

Method	Advantages
ADNet	Mean square error (MSE) refines the training
NERNet	Retains image details and textures when dealing with realistic noise
PCLDCNet	Utilizes ResNet to overcome Network Degradation
MP-DCNN	Incorporating SegNet, it operates end-to-end residual CNN, which extracts noise and reconstructing features of image
DeGAN	Employs GANs, U-Net and VGG-19, which can remove mixed noise
Classifer/regression CNN	Combines classification model and regression model together to reconstruct clean images
CDNet	Complex-valued CNN boosts computational efficiency and outputs real-value image

TABLE 2: Some of CNNs used for general images denoising

In the realm of CNN advancements, several notable implementations and improvements have emerged, each addressing different aspects of image denoising. The Separation Aggregation Network (SANet) utilizes a trio of blocks - the convolutional separation block, deep mapping block, and band aggregation block - to effectively remove noise from images. In contrast, the Detail Retaining CNN (DRCNN) focuses specifically on preserving the integrity of high-frequency image content. Additionally, the Bayesian Deep Matrix Factorization (BDMF) approach has been designed for multi-image denoising, catering to scenarios where multiple images need simultaneous noise reduction[1].

2.3.2 CNNs for specific image denoising

There are also various CNNs that can deal with specific images.

Based on previously mentioned DnCNN, the Spectral-Spatial Denoising Residual Network (SSDRN) represents a significant advancement in CNN-based image denoising. This network, notable for preserving the spectral profile while effectively removing noise, operates as an end-to-end algorithm. It incorporates a three-part structure: spectral difference learning, key band selection, and the denoising process, which is executed through the DnCNN model. In this method, a training set is formed from patch groups, which are then processed using advanced deep learning techniques to efficiently reduce noise[1].

The Two-Stage Cascaded Residual CNN is introduced for the efficient removal of mixed noise from infrared images, integrates a mixed convolutional layer. This layer combines various convolution types, such as dilated, sub-pixel, and standard convolutions, to enhance feature extraction and accuracy. Each network's final convolution layer is equipped with a uniquely designed single filter, optimized for noise reduction in infrared images[12].

To get better use of ResNet and pretraining methods to deal with specific images, a novel approach for despeckling ultrasound images using a pre-trained Residual Learning Network (RLN) is introduced. This method was rigorously tested on both artificial and naturally speckle noise-corrupted images, demonstrating its effectiveness in ultrasound image enhancement[13].

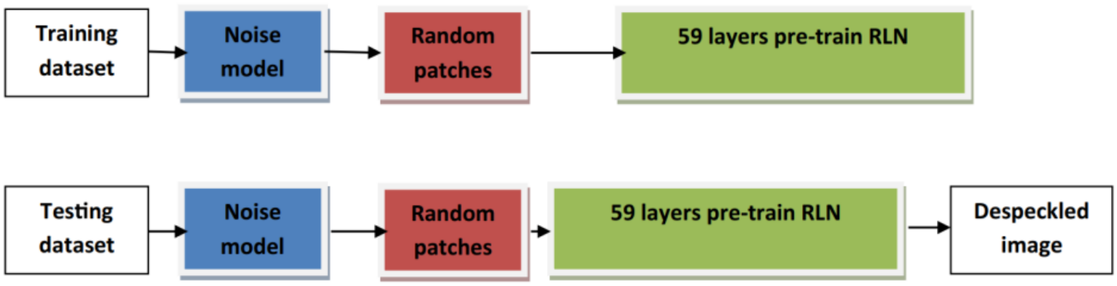


FIGURE 9: Pre-trained RLN [13]

Building upon the previous advancements in image denoising, Progressive Network Learning Strategy (PNLS) tailors for images following the Rician distribution. This approach utilizes large convolutional filters and is structured around two distinct residual blocks. Each block is composed of 5 layers, with three convolution layers positioned strategically between the two blocks to facilitate efficient processing and enhance denoising performance[14].

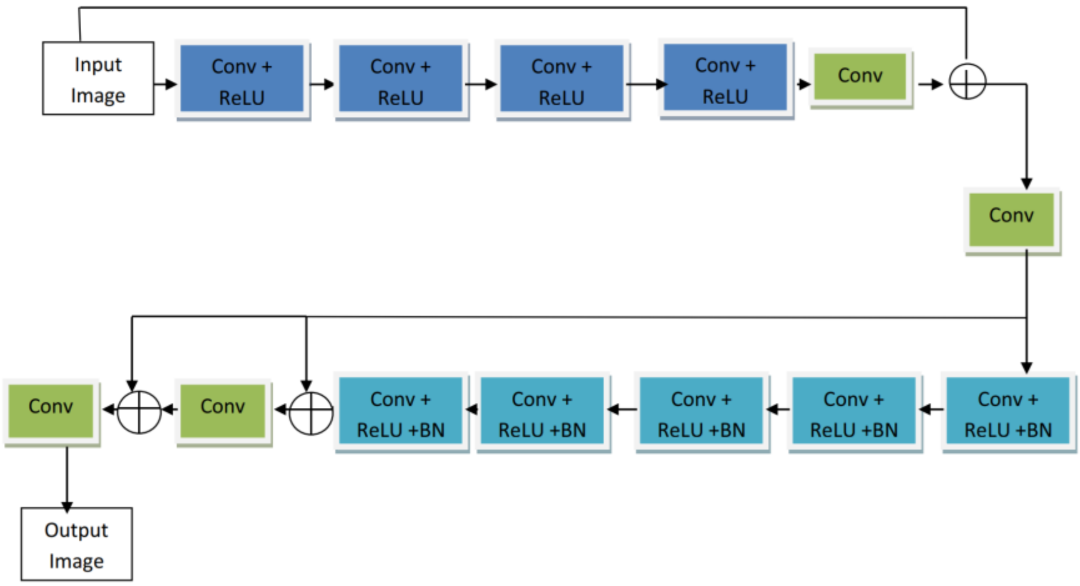


FIGURE 10: PNLS [14]

Methods	Advantages	Application fields
SSDRN	Executes through the DnCNN, using patch groups to form training sets	Spectral profile preservation
Two-phased cascaded residual CNN	Employs residual learning to estimate calibration parameters, uniquely designed single filter	Infrared images denoising
Pre-trained RLN	Comprises both a noise model, created from a training dataset, and the pre-trained RLN itself.	Ultrasound image enhancement
PNLS	Utilizes large convolutional filters and two distinct residual blocks to facilitate efficient processing and enhance denoising performance	Images following the Rician distribution

TABLE 3: Some of CNNs used for specific images denoising

In the realm of specialized image denoising, several innovative methods have been developed for targeted applications. UDnNet, a generative adversarial network, is specifically crafted for denoising underwater images. This network is composed of two main components: a generator network and a discriminator network. Additionally, the Hybrid CNN Method addresses speckle noise reduction, employing a unique strategy. This method starts with networks trained on Gaussian noise models and then fine-tunes them using data that emphasize structural boundaries, effectively combining general noise reduction principles with specific structural considerations. Moreover, the CNN-DMRI represents a breakthrough in MRI scan denoising. This method is built around an encoder-decoder structure, meticulously designed to retain essential image features while efficiently filtering out irrelevant noise components. This tailored approach ensures the preservation of critical details in MRI scans, enhancing the clarity and usability of the resulting images[1].

In the field of image denoising, deep learning approaches can also be divided into supervised or self-supervised categories. This classification is based on whether the methods require pairs of noisy and clean images for training.

2.4 Supervised image denoising

Supervised image denoising is a process where a deep neural network (DNN) is trained using pairs of noisy and corresponding clean images. In this approach, the model is taught to map a noisy image to its clean counterpart. This method can be implemented in two main ways: either by directly learning the transformation from a noisy image to a clean image, as cited in several studies, or by targeting the residual difference between a clean and a noisy image, which effectively teaches the model to separate noise from the actual image content[15].
Once the model is adequately trained, it becomes capable of processing new, unseen noisy images and producing their denoised versions. This capability is particularly valuable in various applications such as medical imaging, astronomical observation, and photography, where maintaining the integrity of the original image is crucial.

Jain et al. pioneered the use of DNNs for image denoising in 2008, achieving results on par with then-leading traditional algorithms like wavelet and MRF methods, but with reduced computational costs. This breakthrough led to the development of more DNN-based methods[16].

Various approaches have been developed to handle different types of noise. Alongside the common Additive White Gaussian Noise (AWGN), noise models like Poisson-Gaussian distribution are also utilized.

In a separate development, Guo et al. proposed the CBDNet, specifically tailored for real-world photography[17] . This network, embracing the Poisson-Gaussian noise model and in-camera processing, consists of two parts: a noise estimation subnetwork and a non-blind denoising subnetwork. The noise estimation subnetwork employs an asymmetric loss function, penalizing underestimation of noise more heavily, thereby improving the robustness of the denoiser.

Following the framework established by CBDNet, a simpler yet effective model SDNet was introduced [18]. It utilizes the generalized signal-dependent noise model and achieves competitive results on both synthetic and real noisy images through a stage-wise process and lifted residual learning. Each of these developments represents a stride forward in the field, offering unique solutions to the complex problem of image denoising.

Besides DNNs, the UNet model, initially established for biomedical image segmentation, has also evolved significantly, finding applications in various domains including image denoising. Its lightweight and high-performance characteristics make it an ideal baseline model for semantic segmentation tasks.

The Dense U-Net (DDUNet)[20] represents an improved version of the original UNet, focusing on applications in image denoising and segmentation. It capitalizes on the strengths of UNet's architecture while enhancing it with additional features to better handle the complexities involved in these tasks.

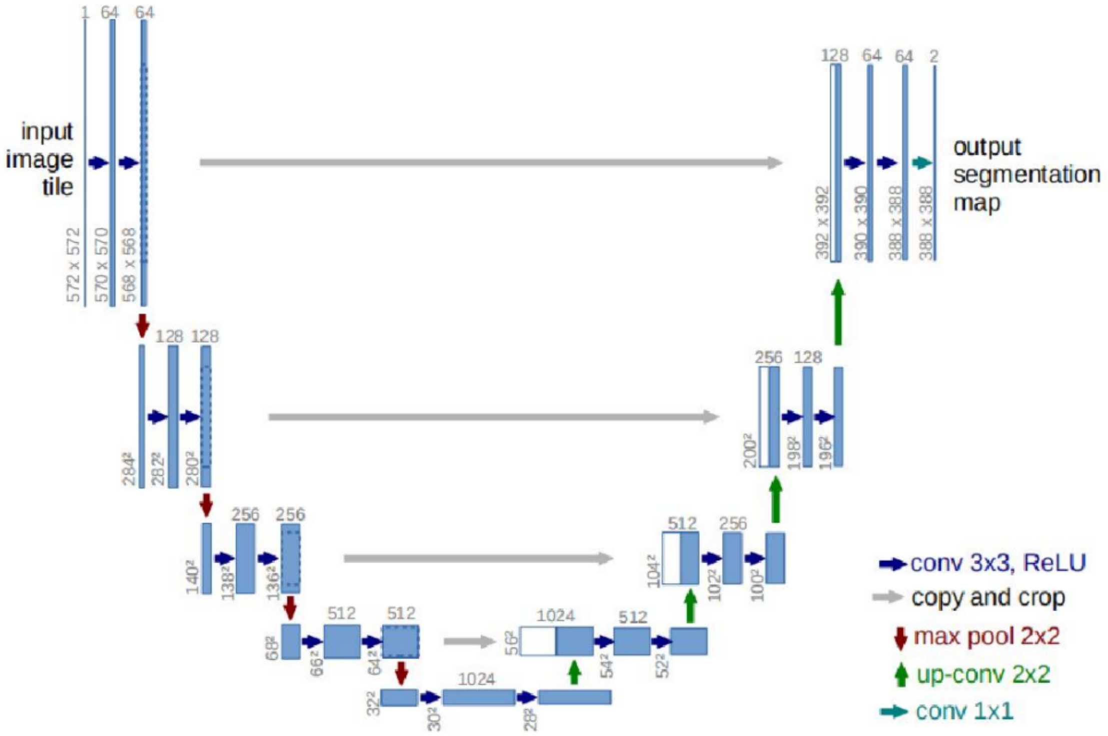


FIGURE 11: the network architecture of UNet [20]

Similarly, the Residual Dense U-Net (RDUNet)[21] introduces densely connected convolutional layers within its encoding and decoding segments. This adaptation aims to leverage the dense connections to capture more intricate image details.

In a shift towards transformer-based models, the Swin Transformer UNet (SUNet)[22] represents a significant departure from traditional CNNs. It combines the strengths of both transformers and CNNs, utilizing shifted windows to reduce computational complexity while maintaining high performance in image restoration tasks.

Lastly, the Multi-Task Attentional U-Net (MTA-Net)[23], designed for Hyperspectral image (HSI) denoising, introduces a unique approach. It features specialized networks and learning strategies that cater to the specific challenges posed by HSI data, including the need for precise noise estimation and effective noise separation.

Each of these UNet variants demonstrates the model's adaptability and effectiveness across different domains, particularly in image denoising and segmentation tasks, highlighting its ongoing evolution and significance in the field of image processing.

Supervised image denoising effectiveness hinges on diverse and quality training data, but acquiring perfect noisy-clean image pairs is challenging. This has led to a shift towards self-supervised methods, which are more flexible as they don't always require clean images. Enhancements in CNNs and GANs further improve these models, allowing for better texture and structure recreation in images. However, the dependency on extensive training data and the practical difficulty of obtaining perfectly clean images in real-world scenarios are significant hurdles. Consequently, researchers often use artificial noise addition to clean datasets, creating synthesized pairs for training, though this doesn't always accurately mimic real-world noise.

2.5 Self-supervised image denoising

Different from supervised algorithm, Self-supervised image denoising algorithm is a deep learning image denoising algorithm that does not require paired noisyclean images as training data.

Blind Spot Network (BSN), a self-supervised image denoising algorithm, operates on the principle that noise is spatially independent and has a zero mean. It leverages the spatial correlation of image signals to predict blind pixels using surrounding pixels. Recently, numerous BSN-based image denoising algorithms have been developed, indicating the method's growing significance in the field.

Additionally, the Transformer, a model known for its ability to extract global information in image processing tasks, has shown to have unique advantages over CNNs. This has led to the categorization of self-supervised image denoising algorithms into three types: General methods, BSN-based methods, and Transformer-based methods. Each type offers distinct approaches and benefits, contributing to the diversity and effectiveness of image denoising techniques in contemporary research.

2.5.1 General self-supervised image denoising methods

The realm of image denoising has witnessed significant advancements through the adoption of self-supervised learning techniques. These methods have diversified the strategies used for reducing noise in images, each with its unique approach and application. This overview presents a variety of general self-supervised image denoising methods, highlighting their respective applications, advantages, and other relevant notes.

Method	Denoising Application	Advantages	Additional Notes
N2N	Gaussian, Poisson, Bernoulli noise denoising and random text overlays remove.	Precisely aligned noisy-noisy image pairs.	Paired noise images are needed
GCBD	Real-world sRGB image noise, Gaussian and Mixture noise denoising.	GAN-generated noise distribution applied to clean images for synthetic noisy-clean pair creation.	Unpaired clean images
SURE-based Method	Gaussian noise	Stein's unbiased risk estimator(SURE) based method for refined risk estimation.	Noise Model
Noisier2noise	Gaussian additive noise and multiplicative Bernoulli noise denoising.	Synthetic noise addition to original noisy images for label generation, followed by applying similar noise types to these labels for input creation.	Arbitrary noise model
Recorrupted to-recorrupted (R2R)	AWGN and real-world sRGB image noise denoising.	Data augmentation technique for generating noisy-noisy pairs through re-corruption.	Noisy level function(NLF) or ISP function
NBR2NBR	Gaussian, Poisson noise and real-world rawRGB image noise denoising.	Creating noisy-noisy pairs by splitting a single noisy image into two sub-noise images.	
Noise2Score	Gaussian, Poisson, and Gamma noise denoising.	Training a Neural Network to estimate the score function and using Tweedie's formula for final denoising is an effective method for handling various exponential family noises.	Arbitrary noise model
NAC	AWGN and real-world	Introducing synthetic noise to	Noise model

Method	Denoising Application	Advantages	Additional Notes
	sRGB image noise denoising.	images with existing weak noise for inputs, using the original weak noise images as targets.	
CVF-SID	Real-world sRGB image noise denoising.	Cyclic multi-Variate Function (CVF) employs a CNN model to split sRGB noise images into clean, signal-independent, and signal-dependent noise components.	
IDR	Gaussian, binomial and impulse noise, real-world raw noise denoising.	Enhancing denoising performance using iterative techniques in the model.	Noise model

TABLE 4: Some General self-supervised image denoising methods

The field of self-supervised image denoising continues to evolve, offering a range of solutions for various types of noise in different imaging contexts. Each method presents a unique set of advantages, whether it is the precision of noisy pair alignment, the generation of synthetic noisy-clean pairs, or the employment of advanced estimators and functions. The diversity of these methods underscores the dynamic nature of image denoising research and its ongoing pursuit to develop more efficient, accurate, and versatile denoising techniques suitable for the challenges of real-world image processing.

2.5.2 BSN-based self-supervised image denoising methods

BSN-based self-supervised image denoising is an innovative method that enhances image quality by predicting noise-free pixels of masked pixels. This technique relies on the spatial continuity between masked pixels and their surrounding counterparts in the image signal. BSN's effectiveness hinges on the premise that image noise is spatially independent and has a zero-mean, whereas the image signal itself demonstrates spatial correlation.

BSN-based methods are primarily categorized into two distinct strategies based on their approach to masking: 'mask in input' and 'mask in network'.

The 'mask in input' approach involves masking certain pixels in the noisy image. This masked image serves as input, with the complete noisy image used as the target for supervised training on deep neural networks.

Method	Noise Type	Applications	Advantages
N2V (Noise2Void)	Gaussian and biomedical	Biomedical imaging	Independently masks random pixels, suitable for pixel-wise noise.
N2S (Noise2Self)	Gaussian (blind)	General	Utilizes J-invariant function for masking, introducing randomness effectively.
PN2V (Probabilistic Noise2Void)	Arbitrary noise models	Microscopy, low-light imaging	Employs probabilistic modeling for accurate intensity prediction.
Noise2Same	Gaussian	General	Adopts J-invariant masking, replaces pixels with local averages for consistency.
S2S (Self2Self)	Gaussian, salt-and-pepper, sRGB	General	Bernoulli sampling creates effective noisy pairs for diverse noise types.
B2UB (Blind2Unblind)	FMDD, Gaussian, Poisson, rawRGB	General	Features global-aware masking and re-visible loss for enhanced denoising.

TABLE 5: BSN-based self-supervised image denoising methods of 'mask in input'

Conversely, the 'mask in network' strategy focuses on masking parts of the receptive field during feature extraction within the network structure. This approach enables the model to use surrounding pixels of the feature maps to predict the target pixel.

Method	Noise Type	Advantages
Laine et al.	Gaussian, Poisson, Impulse	Utilizes masking in four directions to create a blind spot network, enhancing denoising capability.

Method	Noise Type	Advantages
DBSN	AWGN, HG, MG, real-world sRGB, Unpaired clean images	Incorporates dilated convolution, NLF, and knowledge distillation, effective for various noise models.
AP-BSN	Real-world sRGB	Features asymmetric P D in training/testing and a random-replacing refinement post-process.
MM-BSN	Real-world sRGB	Implements a multi-mask strategy for large area noise, enhancing spatial noise reduction.
Li et al.	Real-world sRGB	Differentiates between flat and textured regions, creating tailored supervisions for each.

TABLE 6: BSN-based self-supervised image denoising methods of 'mask in network'

2.5.3 Self-supervised image denoising based on Transformer

Transformers, originally excelling in natural language processing, have also made significant strides in the field of computer vision, including image denoising. However, applying a pure Transformer model directly to self-supervised image denoising can result in sub-optimal outcomes.

To address this, the Context-aware Denoise Transformer (CADT)[24] was developed, enhancing the synergy between Transformer and CNN technologies. It employs a dual-branch structure, combining global and local features, whose approach allows CADT to effectively retain important image details during the denoising process. Figure 2 presents the architecture of Denoise Transformer.

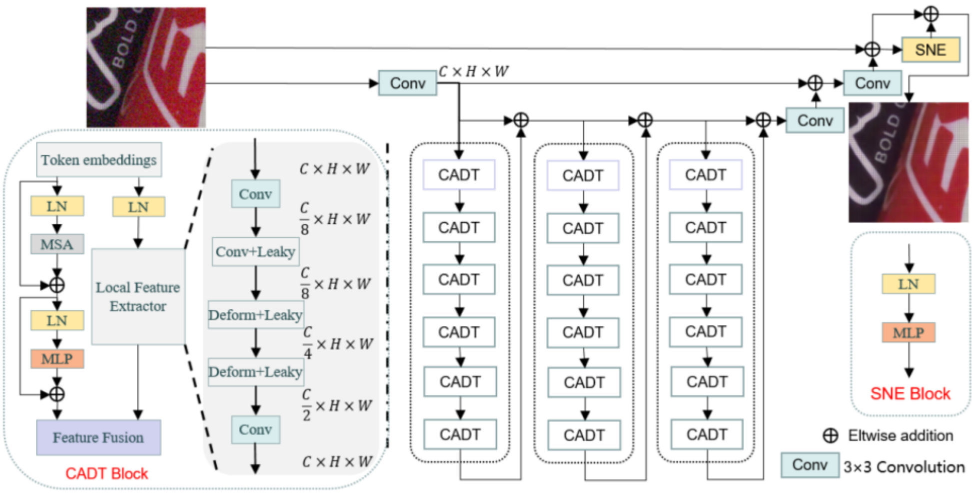


FIGURE 12: The architecture of Denoise Transformer [24]

Additionally, the Cross Transformer Denoising CNN (CTNet)[25] represents another innovative approach in this field. CTNet integrates various structural information from different serial and parallel networks through attention mechanisms. It consists of a serial block (SB), a parallel block (PB), and a residual block (RB), which help convert obtained feature mappings into clean images.

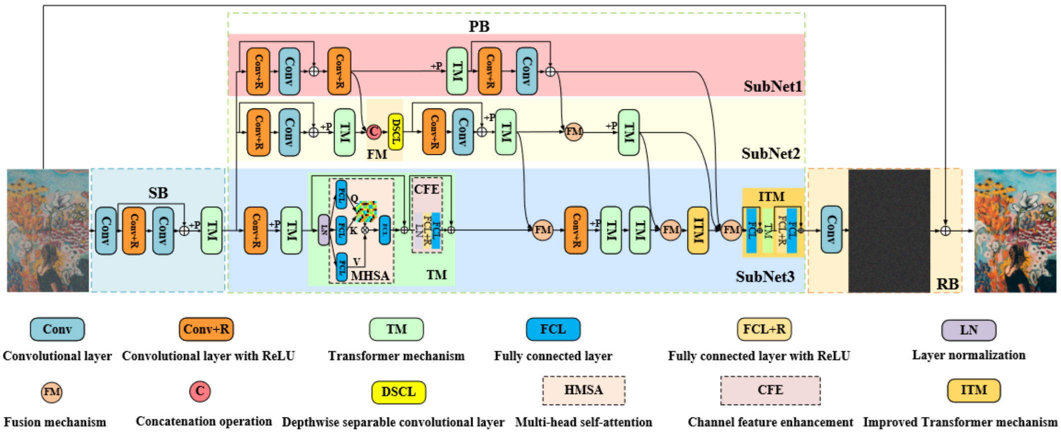


FIGURE 13: Network architecture of CTNet [25]

Method	Mask Way	Applications	Advantages
DT	Mask in inputs	FM dataset, Gaussian, Poisson, and real-world rawRGB images noise denoising	Combines CNN and Transformer for enhanced denoising.
LG-BPN	Mask in networks	Real-world sRGB image noise denoising	Uses DSPMC for blind spot creation and noise break; DTB for integrating local and global information.
SwinIA	Mask in inputs	FM dataset, Gaussian, Poisson, and real-world rawRGB images noise denoising	Employs Transformer for efficient image denoising.

TABLE 7: Some other transformer-based self-supervised image denoising methods.

These developments highlight the ongoing evolution in the field of image denoising, where traditional CNN approaches are being augmented or even replaced by more advanced Transformer-based models, offering improved performance in terms of both efficiency and effectiveness.

2.6 Unsupervised and Application-Specific Denoising

In the domain of application-specific image denoising, different techniques cater to varied and specialized needs across fields. Image denoising is pivotal in enhancing the quality and usability of images in diverse scenarios.

For instance, in medical imaging, especially for detecting diseases such as COVID-19, CT image denoising plays a critical role. Advanced deep learning networks are employed to filter out noise from CT scans, thereby improving the clarity and accuracy of disease detection. This process is crucial for timely and precise diagnosis, which directly impacts patient outcomes[26].

Similarly, in neuroscience, the technique of “Fast, efficient, and accurate neuro-imaging denoising via supervised deep-denoising method” finds application in various settings such as whole-brain imaging, large-field-of-view imaging, and the detailed analysis of complex neurite structures. These applications demand high precision and clarity, given the intricate nature of neuroimaging data[27].

Moving forward, Prior Residual Noise Embedded Denoising Diffusion Probabilistic Models (Resfusion) [28] is a cutting-edge approach that integrates end-to-end models with denoising diffusion models, consisting of a training pipeline and an inference pipeline. It is designed specifically for image segmentation tasks, utilizing the gradual image generation capabilities of diffusion models.

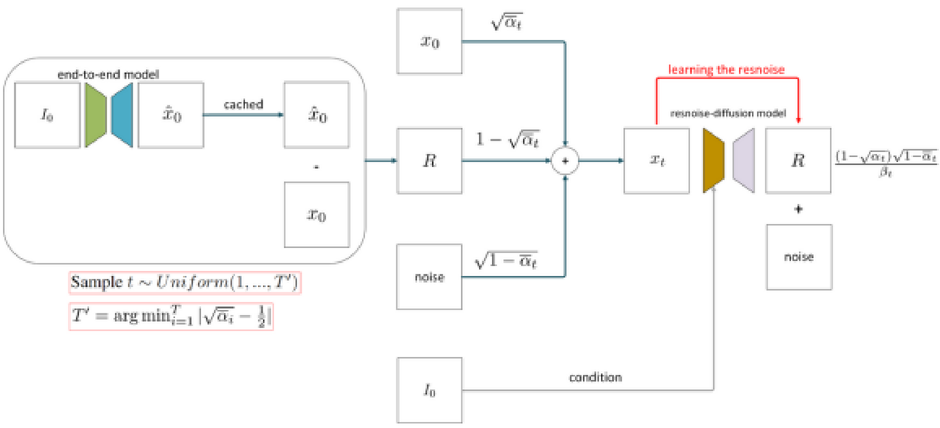


FIGURE 14: The training pipeline of Resfusion [28]

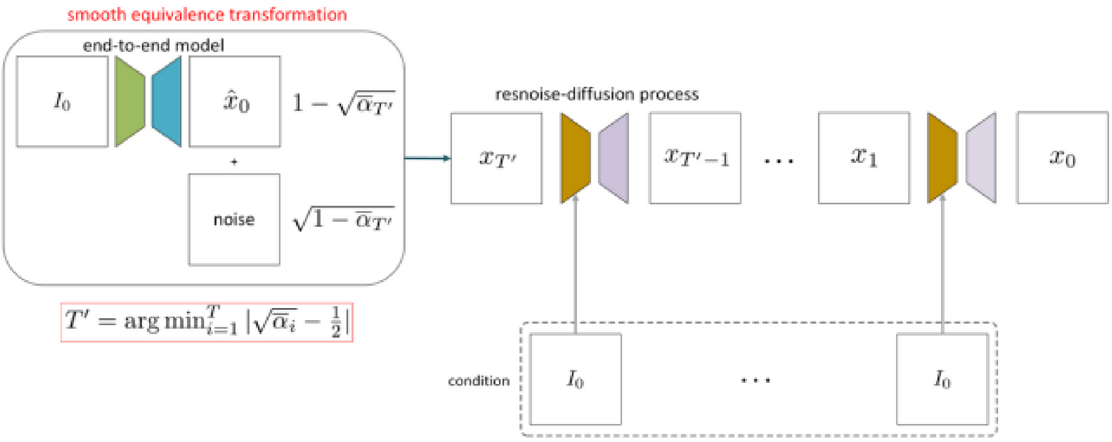


FIGURE 15: The inference pipeline of Resfusion [28]

On the other hand, Unsupervised Image Denoising via Self-Collaboration Parallel Generative Adversarial Branches (SCPGabNet)[29] offers a solution to the challenge of training without paired datasets. SCPGabNet uses a self-collaboration strategy, allowing the network to self-improve without increasing complexity or altering its architecture, showing better results compared to some other denoising methods. This technique is especially beneficial in scenarios, demonstrating the adaptability of denoising methods to real-world challenges.

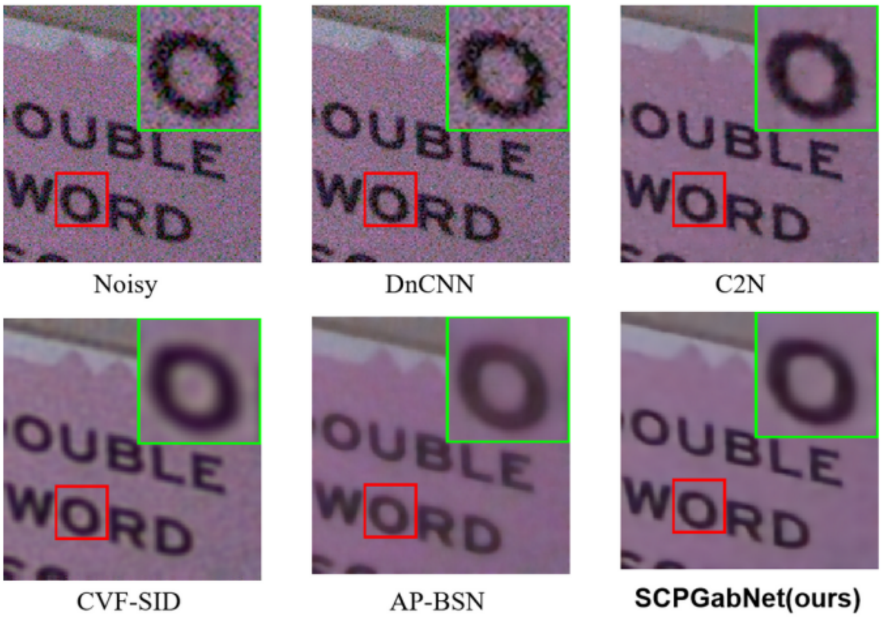


FIGURE 16: A real noisy image from the SSID Validation dataset and results from different methods[29]

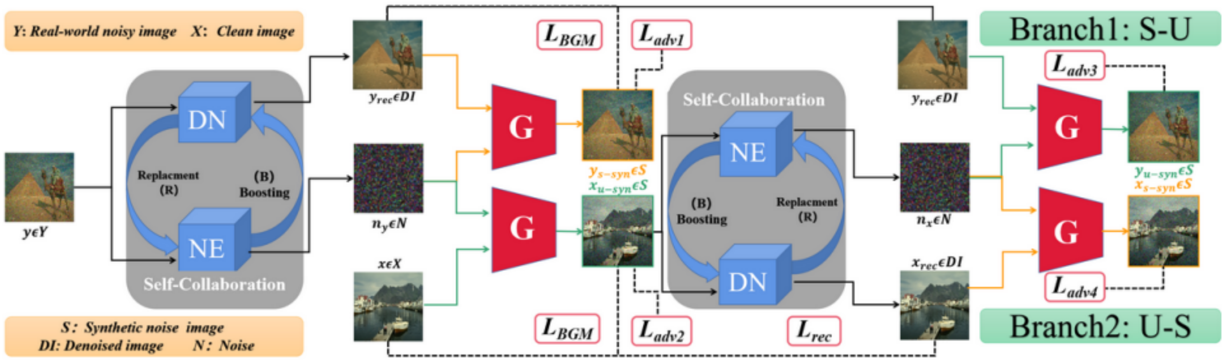


FIGURE 17: The architecture of SCPGabNet framework[29]

Detail Reconstruction in CNN-based Image Denoising Algorithms: Despite advancements, CNN-based image denoising algorithms still face challenges in detail reconstruction, often resulting in over-smoothed images. New approaches like diffusion models and LSTM-enhanced DnCNNs are being explored to preserve high-frequency details and address these shortcomings[30].

End-to-End Fully Unsupervised Denoising Approaches: Fully unsupervised denoising, crucial for scenarios lacking paired training datasets, is gaining traction. Innovative models using GANs and VAEs, like unified end-to-end deep learning models and frequency-sensitive methods, are showing promise in addressing diverse noise types without relying on paired datasets[31].

These studies demonstrate the ongoing evolution in the field of image denoising, leveraging advanced machine learning techniques to address specific challenges in various applications. The integration of diffusion models and generative adversarial networks (GANs) in these methods highlights the trend towards more sophisticated, efficient, and versatile denoising techniques.

Chapter 3: Design and Implementation of CNN

My design and implementation of the improved denoising CNN presents an in-depth examination of digital image noise types and their mitigation through advanced neural network techniques. It explores the characteristics, simulation, and impact of Gaussian, Exponential, Rayleigh, Salt-and-Pepper, Speckle, Poisson, and K-distribution noise with different amplitudes on image quality. Furthermore, enhancements to the original DnCNN architecture aimed at improving noise reduction efficacy and the overall structure of Network are introduced. The process for training models to address various noise types and intensities is also outlined, preparing for robust image denoising results & analysis.

3.1 Datasets with different kinds of noise added

For image prepossessing, the most important part is to delve into the characteristics and simulation of different noise types, including Gaussian, Exponential, Rayleigh, Salt-and-Pepper, Ultrasound Acquisition (Speckle), Quantum (Poisson), and K-distribution noise.

3.1.1 Gaussian Noise

Gaussian noise is a kind of noise that can be found in many types of images and signals. It has a bell-shaped probability distribution, which is also called a "normal distribution" in statistics. Gaussian noise is often used to simulate the random noise that happens in the real world, such as the noise in photographs taken in low light or the staticin a phone call. It can represent many types of real-world noise, including electronic noise in devices and background noise in imaging systems.

The probability density function P of Gaussian distribution is:

$$PG(Z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(Z-\mu)^2}{2\sigma^2}}$$

[32]

3.1.2 Exponential Noise

Exponential noise is a concept closely tied to the exponential distribution, which describes the time between events in a process where events occur continuously and independently at a constant rate. This type of noise is often used in the context of time series forecasting and signal processing, especially when modeling or smoothing data that shows a trend or seasonal patterns.

Typically, the time interval between events in a Poisson process is characterized by the exponential distribution. The probability density function (PDF) for exponential noise is:

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & , x \geq 0, \\ 0 & , x < 0. \end{cases}$$

[32]

3.1.3 Rayleigh Noise

Rayleigh noise is associated with the Rayleigh distribution, a continuous statistical model used for positive variables. This model is particularly relevant in scenarios involving random variables whose magnitude is a combination of two independent, normally distributed components with a mean of zero, and it often occurs in scenarios like wave height modeling in the ocean and background noise in MRI imaging.

The probability density function P of Rayleigh distribution is:

$$PR(Z) = \frac{2}{b}(Z-a)e^{\frac{-(Z-a)^2}{b}} \quad \text{For } Z \geq a$$

$$PR(Z) = 0 \quad \text{Otherwise}$$

[32]

3.1.4 Impulse Noise (Salt & Pepper Noise)

Impulse Noise, also called Salt-and-pepper noise, is a type of noise that appears as sudden disturbances in images, showing up as randomly occurring white and black pixels. It looks like sprinkled white and black dots on the picture, hence the name. This noise can simulate the effect of sharp and sudden disturbances in image data, like errors in data transmission or faulty sensor pixels in cameras.

3.1.5 Ultrasound Acquisition Noise (Speckle Noise)

Ultrasound acquisition noise refers to the unwanted signals or disturbances that occur during the process of capturing ultrasound images. This noise can degrade the quality of the ultrasound image, making it difficult to interpret and analyze. A common type of noise in ultrasound images is speckle noise. Speckle noise is a granular noise that inherently exists in and degrades the quality of ultrasound images. It is caused by the coherent nature of ultrasound imaging, where the interference of sound waves creates a speckled pattern in the image.

In MATLAB, speckle noise is often used to simulate ultrasound acquisition noise because it closely mimics the real-world noise characteristics found in ultrasound images. By using speckle noise in simulations, researchers and engineers can develop and test denoising algorithms in a controlled environment before applying them to actual ultrasound images. This approach allows for the refinement of techniques aimed at improving image quality and the reliability of ultrasound diagnostics.

3.1.6 Quantum Noise (Poisson Noise)

Quantum noise in CT (Computed Tomography) images, also known as photon noise, arises due to the statistical nature of photon detection. In CT imaging, the image quality can be significantly affected by the fluctuations in the number of photons that reach the detectors. Since these photons are quantized particles, their detection follows the principles of quantum mechanics, leading to variations in the recorded signal. This type of noise is inherent in the process of capturing CT images and can limit the ability to distinguish between small or low-contrast objects within the image.

In MATLAB, Poisson noise is commonly used to simulate quantum noise found in CT images. The rationale behind using Poisson noise for simulation lies in the statistical distribution of photon arrival. The number of photons detected follows a Poisson distribution, characterized by the mean number of photons expected to be detected over a certain interval.

3.1.7 K-distribution Noise

The K-distribution noise model simulates scattering in imaging, useful in radar and sonar, by combining multipath scattering (Rayleigh distribution) with thermal noise (exponential distribution). This model is flexible, adjusting to different scenarios by changing its shape (α) and scale (β) parameters. It is widely used in fields like medical imaging and wireless communication where scattering is significant.

The K-distribution's probability density function (PDF) depends on α and β, which control its shape and spread, respectively. A higher α value leads to a distribution closer to Rayleigh, typical for less variable environments, while a higher β value widens the distribution, fitting data from high-intensity scattering areas.

For generating K-distribution noise, the Gamma distribution is used, applying Gamma noise to an image's luminance channel through the formula kNoise = gamrnd(alpha, beta, size(YChannel))

/ (alpha * beta * L);. Here, dividing by (alpha * beta * L) aims to normalize the noise, ensuring that the luminance of the image remains unchanged. This formula normalizes the noise, maintaining the image's luminance. Noise intensity is categorized into high, medium, and low based on α, with equal α and β values. Experiments varied CNN depth for analysis.

3.2 Design and Improvements of Network

The original DnCNN (Deep Convolutional Neural Network for Image Denoising) architecture is a specialized deep learning model designed to reduce noise in images. It starts with an input layer for 256x256 pixel color images, followed by a convolutional layer with a ReLU activation function to extract initial features. It then uses multiple layers combining convolution, batch normalization, and ReLU to learn complex features while ensuring training stability. The architecture concludes with a convolutional layer and a regression layer to predict noise residuals. Typically, DnCNN employs 17 layers for Gaussian noise and 20 layers for other noises, using the MSE loss function for optimization.

To improve the Network, my improvements to the architecture are reflected in the following areas:

3.2.1 Dynamically Adjusted Convolutional Layer Filter Sizes

The filter sizes for each convolutional layer are dynamically adjusted using the linspace function, gradually increasing from 32 to 128. This design may help the network capture image features more effectively at different levels, making the transition from detail to abstraction smoother.

3.2.2 Introduction of Residual Network

At the end of the network, an addition layer is introduced, which adds the output from the previous layer to the original input image. The network learns the noise residual and automatically adds it to the original input image to eventually output the denoised image. This realizes the core idea of residual learning, i.e. directly learning the noise pattern and subtracting it from the input image, helping to preserve more image details, which significantly increases the denoising effects.

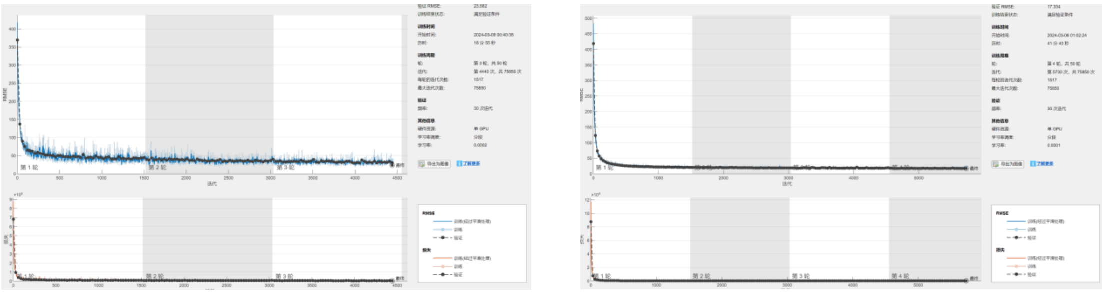


FIGURE 18: Comparation of the training progress between original DnCNN and my improved CNN

As it shown in the graphs of training examples, without the additional layer, the value of RMSE is 23.662, but with the additional later, the value of RMSE is sharply reduced to 17.334.

3.2.3 Customized Loss Layer

By adding a custom loss layer SSIM_MSE_LossLayer, combining Structural Similarity Index (SSIM) and Mean Squared Error (MSE) as the loss function, this combination not only measures pixel-level error but also considers the structure of the image, helping to improve the visual quality of the denoised image.

3.2.4 Fewer layers

It is been found that with fewer layers, the final training results show minimal difference, yet the training time is significantly reduced. This adjustment enhances the model's efficiency without sacrificing much in terms of denoising performance, making it a more practical option for real-world applications.

3.2.5 Key parameters Adjustment

After some testing, the MiniBatchSize, which is the number of data samples processed by the network in each training iteration, was set to 4. This not only ensured the accuracy of model training but also reduced memory usage during the training process, thereby enhancing computational speed and efficiency.

A notable aspect is the initial learning rate set at 1e-4, which is adjusted by halving every 3 epochs under a piecewise schedule. This strategy aids in fine-tuning the model's adjustments to the weights, ensuring gradual learning and improved model convergence and performance.

The setup incorporates L2 regularization to counter overfitting and shuffles the dataset at the beginning of each epoch to prevent learning biases from data order. The training options are designed to be verbose with a visualization of training progress, enhancing the monitoring and analysis of the model's learning.

Validation is conducted with a distinct dataset, evaluated every 30 iterations, with a validation patience of 15, enabling early stopping if no improvement is observed. This concise overview encapsulates the essential training parameters and their rationale, emphasizing the approach's targeted efficiency and effectiveness in neural network training.

3.2.6 Application Specific

The original DnCNN is set to deal with gray-scale images, but my CNN can deal with RGB images.

Based on the types of noise, such as Gaussian noise, salt-and-pepper noise, ultrasound acquisition noise (speckle noise), quantum noise (Poisson noise), and K distribution noise, as well as the different levels of noise intensity categorized as low, medium, and high, models were trained separately. This approach ensures that the model maintains excellent denoising performance across a wide spectrum of noise types and intensities that fit the requirements of specific application fields. By tailoring the training process to the specific characteristics of each noise type and its intensity level, the model can adapt its denoising strategy effectively. This targeted training strategy not only enhances the model’s overall denoising capability but also optimizes its applicability and effectiveness in diverse real-world scenarios, demonstrating its robustness and versatility in handling various noise challenges.

3.3 Structure of Network

Different from the basic DnCNN, my CNN is designed to support RGB 3-channel images. It commences with an image input layer, which is specifically tailored to accommodate the dimensions of the processed images, and is immediately followed by a batch normalization layer. Moreover, the network employs intermediate convolutional (Conv) and batch normalization (BN) layers aimed at efficiently processing and denoising the input noisy images. While Linear rectification function (ReLU) is used as activation function. These layers are critical for learning complex features and patterns inherent in the data. A distinctive feature of this model is the incorporation of an addition layer towards the end. This layer adds the output learned by the network directly to the original input image. By leveraging the advantages of residual learning in this manner, the model simplifies the learning task by focusing on noise residuals rather than directly learning the denoised image. This approach not only facilitates a better understanding of the differences (residuals) between the input and output but also automates the denoising process, thereby producing readily usable denoised images.

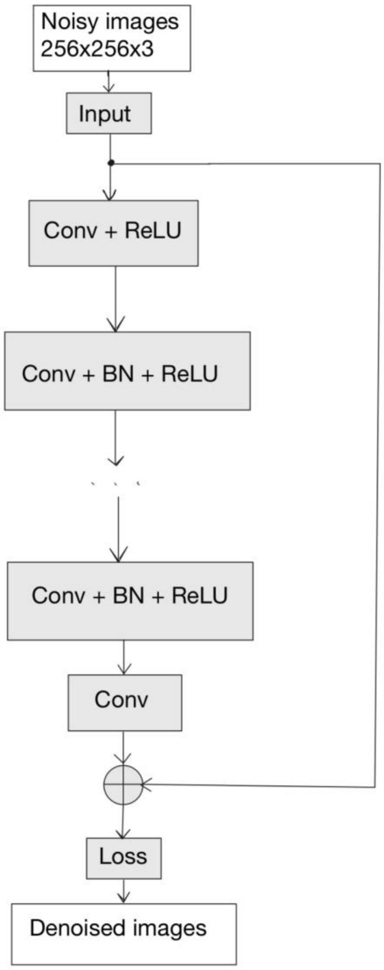


FIGURE 19: The structure of my CNN

3.4 Training progress and code implementation

3.4.1 Image Preprocessing

The implementation initiates by aggregating RGB images from a designated directory, accommodating multiple formats including PNG, JPG, JPEG, and BMP. To ensure uniformity, each image is resized to a standard dimension of 256x256 pixels, enabling a consistent treatment of color information across all images. Additionally, the pixel values of the images are converted to double precision, preparing them for subsequent processing steps in the pipeline.

3.4.2 Noise Addition

Depending on the noiseType parameter, such as gaussian, salt & pepper, speckle, Poisson, K distribution, the corresponding noise with different amplitudes is added to the image datasets. This process generates a set of noisy images alongside the original clean images, which are crucial for the training of the denoising network.

For each image, the variance of the noise added is a randomly generated value, ranging in different level, to ensure diversity in the noise levels processed by the CNN during the training process.

3.4.3 Dataset Splitting

According to Montecarlo principles, the training image dataset is randomly split into a training set (90% of images) and a validation set (10% of images) for each training iteration. This division guarantees that the network is trained on a substantial dataset, while simultaneously being validated on unseen data to monitor and prevent overfitting.

3.4.4 Model Training

The training process employs Montecarlo experiments, where for each iteration, the training and validation datasets are randomly selected, and noise of random intensity within a specified amplitude range is added to each image. This approach ensures the network is trained on noisy images as input with clean images as the target, while validation is performed on a separate, unseen validation dataset. Also, the objective of the training is to adjust the network's weights to minimize the difference between the denoised output and the input images. And the use of residual learning facilitates the network's ability to better learn the residuals between the input and output, enhancing the model's adaptability to randomness in noise conditions. Furthermore, to assess the denoising progress, the loss function is defined as the weighted mean value of the Root Mean Square Error (RMSE) and the Structural Similarity Index Measure (SSIM), which aims to simultaneously optimize the pixel-level accuracy and visual quality of the denoising network, proving beneficial for tasks in image restoration and enhancement.

3.4.5 Model Saving

Post-training, the model and the validation information are saved for later use in denoising new images in “Custom Denoising Networks” file that are ready for further testing and application.

By examining the nature of different amplitudes of noises, my training code sets the stage for the development of targeted denoising strategies. The enhancements to the DnCNN architecture with code training, signify a leap towards more refined and practical denoising solutions.

Chapter 4: Results and Discussion

After training, the effectiveness of denoising models is evaluated using scientific metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). CNN is tested against traditional filters across various noise scenarios, and the influence of adjustments to the model on denoising quality is analyzed. The superior performance of CNN in reducing noise and preserving detail is highlighted, showcasing its adaptability for applications in image processing.

4.1 Evaluation Parameters

To assess the denoising performance of trained models, a more scientific and effective method beyond direct observation of the denoised images involves using specific evaluation parameters. In this project, MSE, PSNR, and SSIM are employed to evaluate the trained model's denoising effectiveness.

4.1.1 MSE

The Mean Squared Error (MSE) is a widely used metric for evaluating the accuracy of predictive models, measuring the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. It is important to note that the L2 loss function is another term for MSE, reflecting the expected value of the squared error loss. MSE can be computed as

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 .$$

The Root Mean Squared Error (RMSE), which is the square root of MSE, is frequently utilized in image denoising to assess the effectiveness of the denoising process. A lower RMSE value indicates a better denoising performance, showing a closer resemblance between the denoised image and the original image.

4.1.2 PSNR

The PSNR (Peak Signal-to-Noise Ratio) is measured in decibels and compares the quality of two images. This metric is frequently employed to assess the quality difference between the original and the processed image. A higher PSNR value indicates improved quality of the resultant image.

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

In this Formula, MAX_i is the maximum possible pixel value of the image, and MSE is Mean Squared Error[33].

4.1.3 SSIM

The Structural Similarity Index Measure (SSIM) is an advanced method used to assess the quality of images by measuring how similar two images are, aligning closely with the perception of quality by the human visual system (HVS). The SSIM approach views any image distortion through the lens of three key components: luminance distortion, contrast distortion, and a reduction in correlation.

SSIM value for two images f and g can be expressed by

$$\mathrm{SSIM} = lfgcfgsfg$$

$$lfg = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1}, \quad cfg = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2}, \quad sfg = \frac{\sigma_{fg} + C_3}{\sigma_f + \sigma_g + C_3}$$

To compute the Structural Similarity Index Measure (SSIM) between two images, f and g, the method evaluates luminance, contrast, and structure through comparison functions l(f,g), c(f,g), and s(f,g), incorporating standard deviations (σ f and σ g), mean values (μ f and μ g), and covariance (σ fg) of the images. Constants C1, C2, and C3 are added to ensure computational stability, preventing division by zero issues. The SSIM index, ranging from 0 to 1, reflects image similarity, with higher values denoting greater resemblance. In image denoising endeavors, SSIM is pivotal for assessing structural fidelity between the denoised output and the original, with higher scores indicating more effective noise removal.[34].

4.2 Plots of Testing Results and Denoising Effects

The network's performance evaluation utilizes a new set of images, the test set, with results derived from a series of Montecarlo experiments conducted five times each, by randomly selecting one-third of the images from the original test set for each experiment, ensuring noise addition consistent with the training phase. Effectiveness is measured using PSNR in dB and SSIM metrics, allowing for direct comparison of the CNN's output against traditional denoising methods like Gaussian, Median, and Bilateral filters. Box diagrams provide a clear, visual representation of the denoising results, enhancing the evaluation with both an intuitive and quantitative analysis. Additionally, the experiments' outcomes are summarized through calculated means and variances of PSNR and SSIM across the five trials, offering a detailed statistical insight into the network's denoising performance.

4.2.1 Gaussian Noise

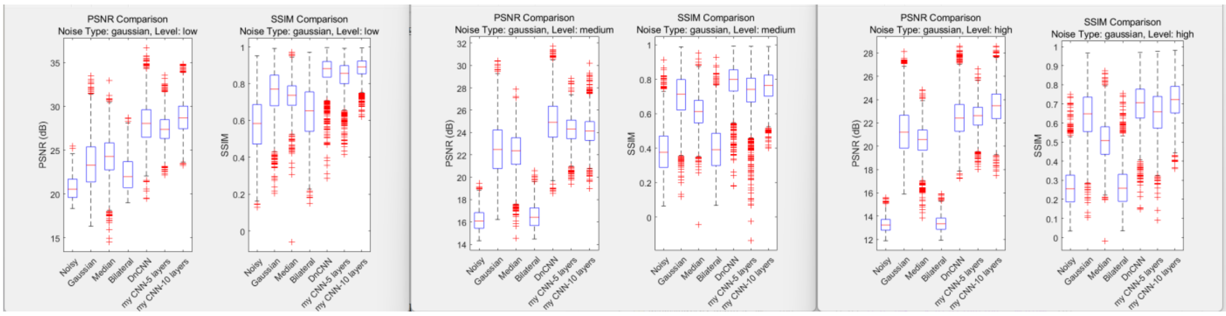


Figure 20: Comparison of PSNR and SSIM for Various Denoising Techniques on Gaussian Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	20.693	23.499	24.278	22.288	28.109	**27.452**	**28.773**
medium	16.164	22.563	22.297	16.515	25.008	**24.248**	**24.167**
high	13.273	21.274	20.478	13.377	22.490	**22.554**	**23.403**

TABLE 8: Comparison of average PSNR for Various Denoising Techniques on Gaussian Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.5731	0.7563	0.7316	0.6401	0.8690	**0.8409**	**0.8847**
medium	0.3823	0.7054	0.6117	0.3963	0.7873	**0.7263**	**0.7601**
high	0.2635	0.6453	0.5081	0.2669	0.6989	**0.6541**	**0.7209**

TABLE 9: Comparison of average SSIM for Various Denoising Techniques on Gaussian Noise-Affected Images at Different Noise Levels

4.2.2 Exponential Noise

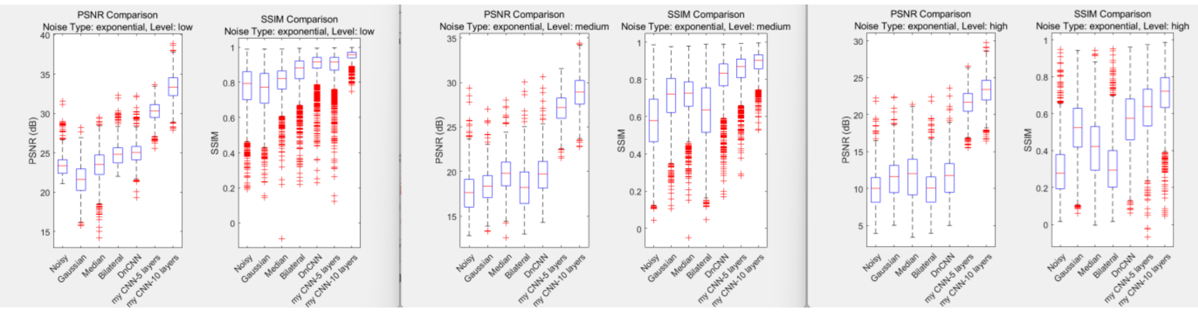


Figure 21: Comparison of PSNR and SSIM for Various Denoising Techniques on Exponential Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	23.280	21.592	23.445	24.696	24.977	**30.263**	**33.372**
medium	17.547	18.355	19.752	18.140	19.618	**27.116**	**28.919**
high	9.853	11.312	11.453	9.913	11.466	**21.604**	**23.302**

TABLE 10: Comparison of average PSNR for Various Denoising Techniques on Exponential Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.7680	0.7539	0.8052	0.8560	0.8984	**0.8902**	**0.9517**
medium	0.5718	0.7064	0.7129	0.6237	0.8097	**0.8481**	**0.8881**
high	0.2941	0.5233	0.4150	0.3109	0.5639	**0.6293**	**0.7058**

TABLE 11: Comparison of average SSIM for Various Denoising Techniques on Exponential Noise-Affected Images at Different Noise Levels

4.2.3 Rayleigh Noise

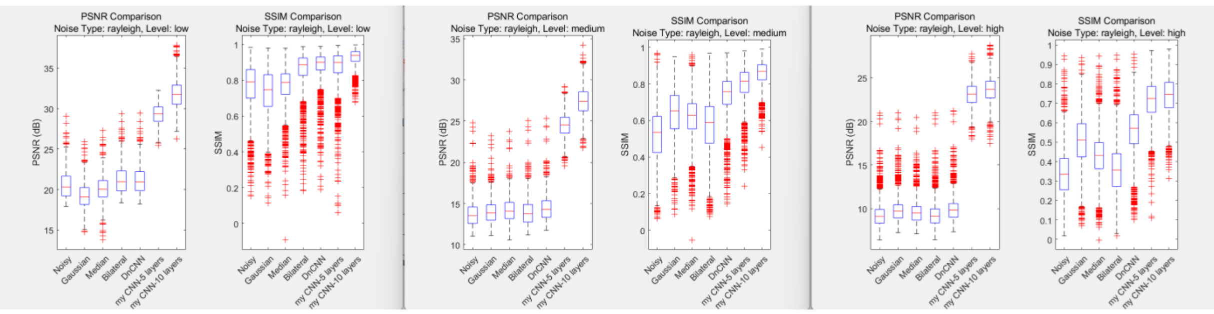


Figure 22: Comparison of PSNR and SSIM for Various Denoising Techniques on Rayleigh Noise-Affected Images at Different Noise Levels

	Noisy	Gaussian	Median	Bilateral	DnCNN	**My CNN**	**My CNN**

	image	filter	filter	filter		**5 layers**	**10 layers**
low	20.471	19.211	20.136	21.105	21.090	**29.290**	**31.806**
medium	13.656	13.974	14.190	13.891	14.416	**24.521**	**27.434**
high	9.253	9.823	9.593	9.298	9.939	**23.096**	**23.649**

TABLE 12: Comparison of average PSNR for Various Denoising Techniques on Rayleigh Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.7666	0.7332	0.7659	0.8589	0.8783	**0.8724**	**0.9273**
medium	0.5216	0.6407	0.6159	0.5707	0.7385	**0.7982**	**0.8569**
high	0.3412	0.5093	0.4314	0.3601	0.5629	**0.7142**	**0.7375**

TABLE 13: Comparison of average SSIM for Various Denoising Techniques on Rayleigh Noise-Affected Images at Different Noise Levels

4.2.4 Impulse Noise (Salt & Pepper Noise)

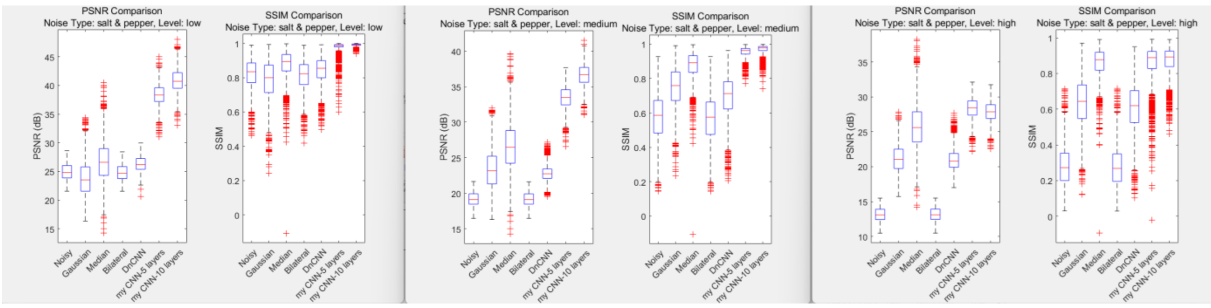


Figure 23: Comparison of PSNR and SSIM for Various Denoising Techniques on Salt & Pepper Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	24.989	23.837	26.796	24.843	26.323	**38.369**	**40.901**
medium	19.186	23.355	26.675	19.185	22.785	**33.455**	**36.672**
high	13.152	21.186	25.820	13.174	20.975	**28.359**	**27.915**

TABLE 14: Comparison of average PSNR for Various Denoising Techniques on Salt & Pepper Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**

low	0.8207	0.7858	0.8770	0.8098	0.8409	**0.9776**	**0.9913**
medium	0.5748	0.7489	0.8749	0.5663	0.6944	**0.9546**	**0.9695**
high	0.2826	0.6412	0.8596	0.2786	0.6117	**0.8571**	**0.8707**

TABLE 15: Comparison of average SSIM for Various Denoising Techniques on Salt & Pepper Noise-Affected Images at Different Noise Levels

4.2.5 Ultrasound Acquisition Noise (Speckle Noise)

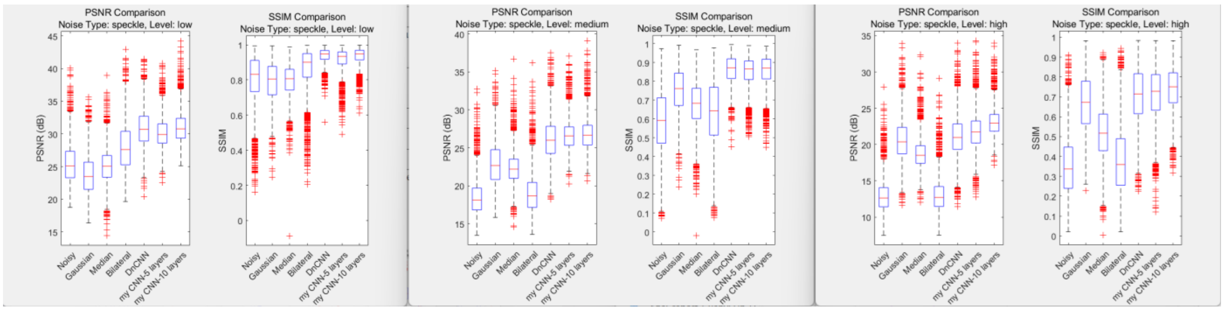


Figure 24: Comparison of PSNR and SSIM for Various Denoising Techniques on Speckle Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	25.516	23.768	25.141	27.989	30.906	**30.201**	**31.085**
medium	18.527	22.939	22.366	19.106	26.178	**26.716**	**26.864**
high	13.002	20.635	18.780	13.127	21.169	**21.817**	**23.155**

TABLE 16: Comparison of average PSNR for Various Denoising Techniques on Speckle Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.8003	0.7894	0.7965	0.8613	0.9378	**0.9138**	**0.9350**
medium	0.5845	0.7545	0.4746	0.6293	0.8632	**0.8499**	**0.8541**
high	0.3542	0.6720	0.5172	0.3808	0.7050	**0.7094**	**0.7376**

TABLE 17: Comparison of average SSIM for Various Denoising Techniques on Speckle Noise-Affected Images at Different Noise Levels

4.2.6 Poisson Noise

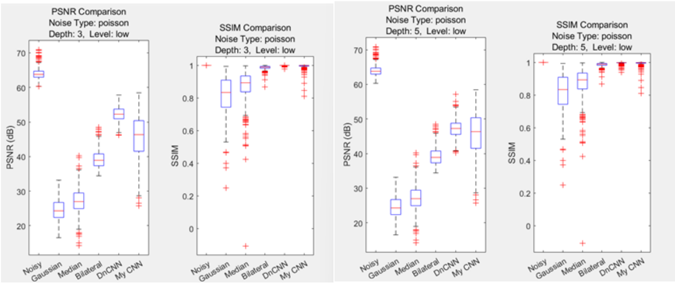


Figure 25: Comparison of PSNR and SSIM for Various Denoising Techniques on Poisson Noise-Affected Images at Different Noise Levels

Notably, for Poisson noise, these denoising methods prove inapplicable.

4.2.7 K-Distribution Noise

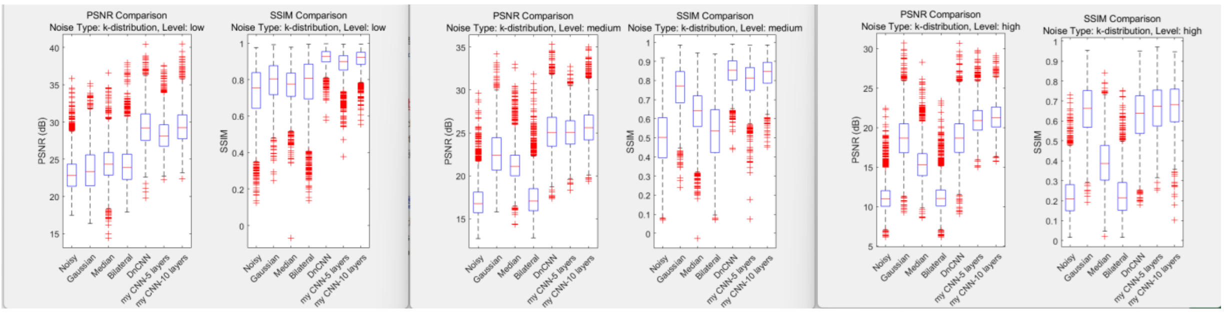


Figure 26: Comparison of PSNR and SSIM for Various Denoising Techniques on K-Distribution Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	23.073	23.651	24.411	24.199	29.389	**28.306**	**29.481**
medium	17.145	22.641	21.333	17.502	25.214	**25.100**	**25.741**
high	11.213	18.725	15.426	11.287	18.715	**20.960**	**21.365**

TABLE 18: Comparison of average PSNR for Various Denoising Techniques on K-Distribution Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.7266	0.7891	0.7654	0.7742	0.9207	**0.8832**	**0.9106**
medium	0.4954	0.7604	0.6317	0.5295	0.8466	**0.8006**	**0.8364**
high	0.2225	0.6587	0.3933	0.2306	0.6295	**0.6643**	**0.6736**

TABLE 19: Comparison of average SSIM for Various Denoising Techniques on Gaussian Noise-Affected Images at Different Noise Levels

4.2.8 Maximum Variances at Noise Intensity across Various Types of Noise and CNN

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.01034	0.02069	0.01703	0.01105	0.00926	**0.00802**	**0.01292**
medium	0.02007	0.02173	0.00933	0.02309	0.01311	**0.00571**	**0.00635**
high	0.03149	0.02711	0.02217	0.03175	0.02779	**0.00416**	**0.00561**

TABLE 20: Comparison of max variance of PSNR across various types of noise & corresponding denoising CNN at different noise amplitudes

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	5.033e-5	6.189e-5	1.977e-5	6.158e-5	3.555e-5	**2.369e-5**	**8.129e-6**
medium	0.00010	6.190e-5	5.660e-5	0.000121	5.064e-5	**2.404e-5**	**6.418e-5**
high	9.047e-5	8.320e-5	9.349e-5	0.000110	7.666e-5	**7.718e-5**	**5.275e-5**

TABLE 21: Comparison of max variance of SSIM across various types of noise & corresponding denoising CNN at different noise amplitudes

The variability of the results from these experiments is quantified as variance. Notably, the variances across all methods, especially my CNN, are relatively low and consistent, showcasing minimal fluctuations across multiple trials. This stability is further highlighted by presenting only the maximum variance values for each type of noise and the corresponding denoising CNN at various noise amplitudes. Detailed results are provided in the Appendix, underscoring the robustness and reliability of the outcomes.

4.3 Discussions and Analysis

When comparing the effectiveness of different network depths, my 5 layer models are superior in denoising performance and 10 layers models even outperforms without worrying the problems like overfitting, while traditional filters and DnCNN with more layers are less efficient. The streamlined models are beneficial for systems with lower computational capacity, offering robust denoising and faster training convergence.

A direct comparison with the original DnCNN model shows that my modified CNN achieves and even often surpasses the original's performance, providing enhanced noise reduction across various conditions and requiring less training time and computational resources.

Overall, my experiment demonstrates that the enhanced CNN model outperforms traditional filters and the basic DnCNN in noise reduction, showcasing exceptional performance in high-amplitude noise situations and detail preservation in color images. It effectively reduces low to medium noise levels and remains robust under heavy noise. The incorporation of an additional layer has markedly improved its denoising capability. Utilizing a combined MSE and SSIM loss function has significantly elevated PSNR and SSIM scores, signifying improved image quality and fidelity.

4.4 Application of my Denoising Neural Networks

Image denoising is critical in image processing, enhancing images by removing noise and preserving details, aiding subsequent tasks such as classification. My CNN, offered in both 5-layer and 10-layer versions, caters to efficiency or performance needs. Users can choose a pre-trained model or train their own based on noise type and computational capacity. The 5-layer model requires less computational effort, trains faster, and surpasses traditional filters and DnCNN, while the 10-layer variant, though more resource-intensive, delivers unmatched outcomes, providing flexibility for varied scenarios.

For example, the denoising CNN I have developed can improve image classification accuracy, as demonstrated by its application in distinguishing bees from wasps, with the performance visualized through Receiver Operating Characteristic (ROC) curves, with the Area Under Curve (AUC) metric providing a comprehensive evaluation.

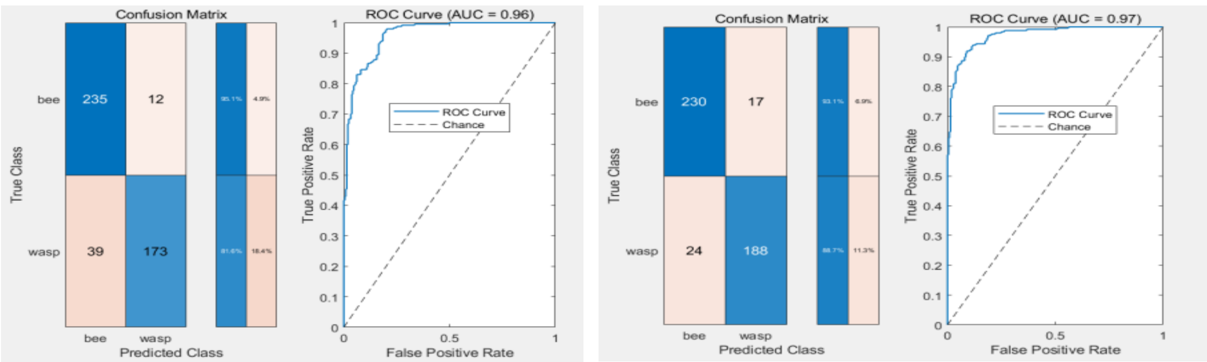


FIGURE 27: Classification results with noise (left) and after denoising (right).

After denoising noisy images using my CNN, there is a notable improvement in classification accuracy, which increases from 88.89% to 91.06%, and the AUC improves from 0.96 to 0.97. This underscores the effectiveness of the denoising process in enhancing the precision of image classification tasks.

Chapter 5: Conclusion and Further Work

5.1 Conclusion

In conclusion, my project embarked on an exploration of advanced image denoising techniques using Convolutional Neural Networks (CNN), with the primary goal of enhancing image classification accuracy. Through the meticulous application of CNN, the project effectively addressed various types of noise that commonly degrade image quality, including Gaussian, salt-and-pepper, speckle, and K-Distribution noise, among others. The use of MATLAB as the platform for implementation enabled the leverage of its advanced computational efficiency and functionality, which significantly contributed to the project's success in developing a comprehensive solution for image denoising.

The outcomes of the project complete all expected work plan. Key achievements include demonstrating CNN's superiority over traditional methods by exploring various denoising approaches in Chapter 2, code design and implementation with novel modifications detailed in Chapter 3, and validating the improved models' effectiveness across noise types with different evaluation parameters like PSNR and SSIM to analyze the results and in Chapter 4. These advancements not only showcased various CNN's robust performance, especially my improved version of CNN, in different areas, but also help solve the problems in image denoising, contributing to the fields of image processing like classification.

In essence, this project has made a contribution to the field of image denoising and classification. By addressing noise types and different amplitudes, with utilizing MATLAB for its computational efficiency, my project developed a comprehensive denoising solution.

5.2 Reflection

5.2.1 Technical Skills Developed

Throughout this project, I significantly enhanced my proficiency in applying Convolutional Neural Networks (CNNs) to the complex task of image denoising, leveraging MATLAB as a powerful tool for both implementation and analysis. The development of CNN architectures specifically tailored for various noise types in digital images allowed me to gain practical experience in neural network customization, model optimization, and data handling. Moreover, designing rigorous experiments to evaluate these models honed my analytical skills. This not only improved my technical abilities in deep learning and MATLAB programming but also enriched my problem-solving toolkit such as the use of generative AI, enabling me to address and overcome computational and algorithmic challenges efficiently.

5.2.2 New Knowledge Gained

The project deepened my understanding of the theoretical foundations and practical applications of deep learning in the context of image denoising. It illuminated the advantages of CNN-based methods over traditional approaches, particularly in their ability to handle diverse noise conditions with superior results. This exploration into the latest developments in CNN architectures for image denoising expanded my knowledge of how to effectively apply these models in real-world scenarios, balancing considerations of model performance, computational efficiency, and application-specific requirements. Furthermore, the project provided insights into the ethical, environmental, and practical considerations of deploying advanced deep learning techniques, underscoring the importance of responsible innovation in the field of digital image processing.

5.2.3 Lessons Learnt on the Project Journey

On my project journey, I honed my skills in remote communication and self-directed learning during the initial phase at BUPT, where I leveraged online resources like Bilibili for foundational knowledge in image denoising. Transitioning to UPV in Valencia, Spain, I immersed in a new academic culture, advancing the project with continuous mentor exchanges and expanding my expertise into practical applications and advanced deep learning concepts, such as attention mechanisms.

In discussions with my mentor, I also understood that our project steered clear of ethical, social, legal, and environmental issues, primarily because we utilized openly available online datasets for training purposes. This aspect ensured our research integrity and aligned with global standards for responsible academic practice.

This journey has been instrumental in developing my adaptability to diverse learning environments and enhancing my ability to self-learn and communicate effectively across cultural boundaries. The project not only advanced my technical knowledge and skills but also provided a valuable perspective on the importance of open-mindedness and perseverance in research.

5.3 Further work

5.3.1 PyTorch implementation and cloud computing

Utilizing PyTorch, an advanced Python machine learning library, for coding facilitates the deployment on cloud computing platforms such as Google Colab to surmount computational power limitations. This approach lays the foundation for researching more complex algorithms and conducting larger-scale, deeper image denoising training.

5.3.2 Exploration of mode advanced methods

Considering the challenges in obtaining clean-noisy image pairs for training, further work could explore unsupervised and semi-supervised learning paradigms and even zero-shot and large model based algorithms. These approaches could leverage unpaired or limited labeled data, making the training process more practical for real-world applications where clean reference images are not readily available.

5.3.3 Improvements on application

Optimizing the proposed model for more efficient in more image processing fields would make advanced denoising capabilities more accessible. This involves model compression techniques, quantization for more application fields like image classification, which could also involve implementing user-guided noise profiling and selective denoising features.

By addressing these areas, my future work can extend the capabilities of image denoising technologies, enhancing both the theoretical understanding and practical applications of deep learning in digital image processing.

NOTE: The maximum length of the report up to here is 50 pages.

References

[1] Methods for image denoising using convolutional neural network: a review. Ademola E. Ilesanmi1,2 · Taiwo O. Ilesanmi

[2] Goyal B, Dogra A, Agrawal S, Sohi BS, Sharma A (2020) Image denoising review: from classical to state-of-the-art approaches. Inform Fusion 55:220–244

[3] Fan L, Zhang F, Fan H et al (2019) Brief review of image denoising techniques. Vis Comput Ind Biomed Art 2:7. https:// doi.org/10.1186/s42492-019-0016-7

[4] Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classifcation with deep convolutional neural networks. In: Advances in Neural information Processing Systems, pp 1097–1105

[5] Deep Residual Learning for Image Recognition. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun(2015)

[6] Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Trans Image Process 26(7):3142–3155

[7] Schwenker F, Kestler HA, Palm G (2001) Three learning phases for radial-basis-function networks. Neural Netw 14(4):439–458

[8] Gai S, Bao Z (2019) New image denoising algorithm via improved deep convolutional neural network with perceptive loss. Expert Syst Appl 138:112815

[9] Lyu Q, Guo M, Pei Z (2020) DeGAN: mixed noise removal via generative adversarial networks. Appl Soft Comp J 95:106478

[10] Jin L, Zhang W, Ma G, Song E (2019) Learning deep CNNs for impulse noise removal in images. J Vis Commun Image R 62:193–205

[11] Quan Y, Chen Y, Shao Y, Teng H, Xu Y, Ji H (2021) Image denoising using complex-valued

deep CNN. Pattern Recogn 111:107639

[12] Guan J, Lai R, Xiong A, Liu Z, Gu L (2020) Fixed pattern noise reduction for infrared images based on cascade residual attention CNN. Neurocomputing 377:301–313

[13] Kokil P, Sudharson S (2020) Despeckling of clinical ultrasound images using deep residual learning. Comp Methods Programs Biomed 194:105477

[14] Li S, Zhou J, Liang D, Liu Q (2020) MRI denoising using progressively distribution-based neural network. Magn Reson Imaging 71:55–68

[15] D. Zhang, F. Zhou, X. Yang, and Y. Gu, "Unleashing the Power of Self-Supervised Image Denoising: A Comprehensive Review," arXiv:2308.00247v3 [eess.IV], Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.00247.

[16] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. Advances in Neural Information Processing Systems, 21, 2008.

[17] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei 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

[18] Hengyuan Zhao, Wenze Shao, Bingkun Bao, and Haibo Li. A simple and robust deep convolutional approach to blind image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pages 0–0, 2019

[19] M. Elad, B. Kawar, and G. Vaksman, "Image Denoising: The Deep Learning Revolution and Beyond—A Survey Paper," 2023. [Online]. Available: https://doi.org/10.1137/23M1545859.

[20] J. Cheng, S. Tian, L. Yu, S. Liu, C. Wang, Y. Ren, H. Lu, and M. Zhu, "DDU-Net: A dual dense U-structure network for medical image segmentation," Applied Soft Computing ,2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1568494622004860?dgcid=coauthor.

[21] J. Gurrola-Ramos, O. Dalmau, and T. E. Alarcón, "A Residual Dense U-Net Neural Network for Image Denoising," IEEE, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9360532.

[22] C.-M. Fan, T.-J. Liu, and K.-H. Liu, "SUNet: Swin Transformer UNet for Image Denoising," 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9937486/authors#authors.

[23] F. Xiong, Z. Gu, W. Zheng, T. Li, and J. Zhou, "Multi-Task Attentional U-Net for Hyperspectral Image Denoising," School of Computer Science and Engineering, Nanjing University of Science and Technology, China, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10283365.

[24] D. Zhang and F. Zhou, "Self-Supervised Image Denoising for Real-World Images with Context-aware Transformer," 2023.

[25] C. Tian, M. Zheng, W. Zuo, S. Zhang, Y. Zhang, and C.-W. Ling, "A cross Transformer for image denoising," 2023. [Online]. Available: https://arxiv.org/abs/2310.10408.

[26] S. U. Khan, I. Ullah, N. Ullah, S. Shah, M. El Affendi, and B. Lee, "A novel CT image de-noising and fusion based deep learning network to screen for disease (COVID-19)," Nature, 2023. [Online]. Available: https://www.nature.com/articles/s41598-023-33614-0#:~:text=,Nature%E2%80%A0www.nature.com%E3%80%91.

[27] S. Chaudhary, S. Moon, and H. Lu, "Fast, efficient, and accurate neuro-imaging denoising via supervised deep learning," Nature, 2023. [Online]. Available: https://www.nature.com/articles/s41467-022-32886-w#article-comments.

[28] S. Zhenning, D. Changsheng, P. Bin, X. Xueshuo, H. Along, Q. Qiaoying, and L. Tao, "Resfusion: Prior Residual Noise embedded Denoising Diffusion Probabilistic Models," 2023. [Online]. Available: https://arxiv.org/abs/2311.14900. [Accessed: 25 Nov 2023].

[29] X. Lin, C. Ren, X. Liu, J. Huang, and Y. Lei, "Unsupervised Image Denoising in Real-World Scenarios via Self-Collaboration Parallel Generative Adversarial Branches," 2023. [Online].

Available: https://ar5iv.labs.arxiv.org/html/2308.06776. [Accessed: 13 Aug 2023].

[30] J. Zhang, J. Zhang, and H. Hong, "Image denoising based on improved RDN algorithm," IEEE, January 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9994352#:~:text=,January%202023%20ISBN%20Information.

[31] V. C. Dodda, L. Kuruguntla, K. Elumalai, S. Chinnadurai, J. T. Sheridan, and I. Muniraj, "A denoising framework for 3D and 2D imaging techniques based on photon detection statistics," Nature, 2023. [Online]. Available: https://www.nature.com/articles/s41598-023-27852-5.

[32] S. Chavan and N. S. Choubey, "Review on Various Noise Models and Image Restoration Techniques," Mukesh Patel School of Technology Management and Engineering, Shirpur Campus, [Online]. Available: https://www.journalijdr.com/sites/default/files/issue-pdf/9981.pdf.

[33] D. J. Hemanth, D. Gupta, V. E. Balas, Eds., Intelligent Data Analysis for Biomedical Applications: Challenges and Solutions, Intelligent Data-Centric Systems, 2019. [Online]. Available: https://www.sciencedirect.com/book/9780128155530/intelligent-data-analysis-for-biomedical-applications

[34] S. Mahmoudpour, M. Kim, "Emerging Trends in Image Processing, Computer Vision and Pattern Recognition," L. Deligiannidis, H. R. Arabnia, Eds., 2015. [Online]. Available: https://www.sciencedirect.com/book/9780128020456/emerging-trends-in-image-processing-computer-vision-and-pattern-recognition

Acknowledgement

First and foremost, I would like to express my gratitude to my supervisor. Before I arrived at UPV for my exchange program, he was actively communicating with me via email, providing guidance on the project, explaining the project's implementation direction in detail, and areas where I needed improvement. After my arrival at UPV, he patiently guided me through my code and thesis writing, enabling me to successfully complete the project. In addition, he also offered invaluable advice on my daily life in Valencia, Spain, as well as on maintaining a balance between study and life, from which I benefited greatly.

Secondly, I would like to thank Dr.Cindy, who guided us in making the necessary preparations before we went to Valencia, such as visa materials. She even came to Valencia to personally organize mock defenses and offer guidance, playing a significant role in enhancing our learning capabilities.

Thirdly, I am very grateful to my parents. Since they provided me with not only the financial support necessary for living expenses in Spain during my final project but also the funds to travel across Europe, enhancing my educational experience with cultural richness and diversity. Their unwavering support and care helped me navigate the challenges of living and studying abroad, making this journey not just possible but deeply rewarding.

Lastly, my journey would not have been the same without the companionship and support of Longwei Xiao, Cenghua Yu, Weiyao Li, and Yunfu Ao. Together, we ventured to Spain, embarking on a journey filled with learning and exploration. The bond we formed during this period was not merely out of convenience but forged from shared experiences and mutual support, making our time in Spain all the more memorable and enriching.

Appendices

Disclaimer

This report is submitted as part requirement for the undergraduate degree programme at Queen Mary University of London, and Beijing University of Posts and Telecommunications. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

BUPT No.: 2020213069
QM No.: 200977962
Full Name (Pin Yin): Guanyi Lin
Full Name (Chinese): 林冠逸

Signature:

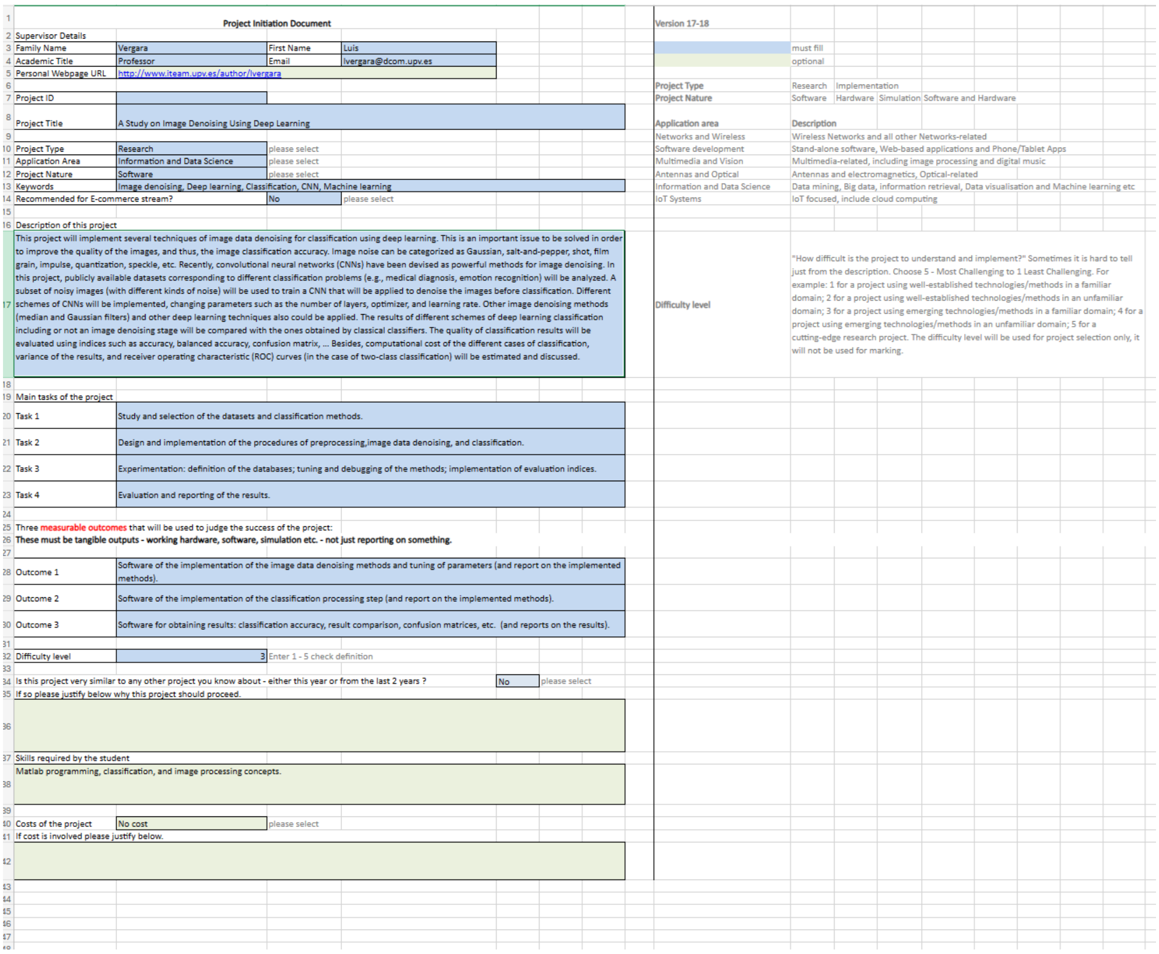
Guanyi Lin
林冠逸

Date:24-04-2024

Project specification

Include your project specification, part 1 and part 2 here. It must be the final version submitted to QMPlus.

Part 1



Project Initiation Document

Supervisor Details

Family Name	Vergara	First Name	Luis
Academic Title	Professor	Email	lvergara@dcom.upv.es
Personal Webpage URL	http://www.iteam.upv.es/author/lvergara		

Project ID		
Project Title	A Study on Image Denoising Using Deep Learning	
Project Type	Research	please select
Application Area	Information and Data Science	please select
Project Nature	Software	please select
Keywords	Image denoising, Deep learning, Classification, CNN, Machine learning	
Recommended for E-commerce stream?	No	please select

Description of this project

This project will implement several techniques of image data denoising for classification using deep learning. This is an important issue to be solved in order to improve the quality of the images, and thus, the image classification accuracy. Image noise can be categorized as Gaussian, salt-and-pepper, shot, film grain, impulse, quantization, speckle, etc. Recently, convolutional neural networks (CNNs) have been devised as powerful methods for image denoising. In this project, publicly available datasets corresponding to different classification problems (e.g., medical diagnosis, emotion recognition) will be analyzed. A subset of noisy images (with different kinds of noise) will be used to train a CNN that will be applied to denoise the images before classification. Different schemes of CNNs will be implemented, changing parameters such as the number of layers, optimizer, and learning rate. Other image denoising methods (median and Gaussian filters) and other deep learning techniques also could be applied. The results of different schemes of deep learning classification including or not an image denoising stage will be compared with the ones obtained by classical classifiers. The quality of classification results will be evaluated using indices such as accuracy, balanced accuracy, confusion matrix, … Besides, computational cost of the different cases of classification, variance of the results, and receiver operating characteristic (ROC) curves (in the case of two-class classification) will be estimated and discussed.

Main tasks of the project

Task 1	Study and selection of the datasets and classification methods.
Task 2	Design and implementation of the procedures of preprocessing,image data denoising, and classification.
Task 3	Experimentation: definition of the databases; tuning and debugging of the methods; implementation of evaluation indices.
Task 4	Evaluation and reporting of the results.

Three measurable outcomes that will be used to judge the success of the project:

These must be tangible outputs - working hardware, software, simulation etc. - not just reporting on something.

Outcome 1	Software of the implementation of the image data denoising methods and tuning of parameters (and report on the implemented methods).
Outcome 2	Software of the implementation of the classification processing step (and report on the implemented methods).
Outcome 3	Software for obtaining results: classification accuracy, result comparison, confusion matrices, etc. (and reports on the results).

Difficulty level	3	Enter 1 - 5 check definition

Is this project very similar to any other project you know about - either this year or from the last 2 years ? No please select

If so please justify below why this project should proceed.

Skills required by the student

Matlab programming, classification, and image processing concepts.

Costs of the project	No cost	please select

If cost is involved please justify below.

Version 17-18

must fill

optional

Project Type	Research	Implementation		
Project Nature	Software	Hardware	Simulation	Software and Hardware

Application area	Description
Networks and Wireless	Wireless Networks and all other Networks-related
Software development	Stand-alone software, Web-based applications and Phone/Tablet Apps
Multimedia and Vision	Multimedia-related, including image processing and digital music
Antennas and Optical	Antennas and electromagnetics, Optical-related
Information and Data Science	Data mining, Big data, information retrieval, Data visualisation and Machine learning etc
IoT Systems	IoT focused, include cloud computing

Difficulty level

"How difficult is the project to understand and implement?" Sometimes it is hard to tell just from the description. Choose 5 - Most Challenging to 1 Least Challenging. For example: 1 for a project using well-established technologies/methods in a familiar domain; 2 for a project using well-established technologies/methods in an unfamiliar domain; 3 for a project using emerging technologies/methods in a familiar domain; 4 for a project using emerging technologies/methods in an unfamiliar domain; 5 for a cutting-edge research project. The difficulty level will be used for project selection only, it will not be used for marking.

北京邮电大学 本科毕业设计（论文）任务书

Project Specification Form

Part 2 - Student

学院 **School**	International School	专业 **Programme**	**Telecommunications Engineering with Management**		
姓 **Family name**	林 Lin	名 **First Name**	冠逸 Guanyi		
BUPT 学号 **BUPT number**	2020213069	**QM 学号** **QM number**	200977962	**班级** **Class**	2020215102
论文题目 **Project Title**	A Study on Image Denoising Using Deep Learning				

论文概述 **Project outline** **Write about 500-800 words** **Please refer to Project Student Handbook section 3.2**	**Understanding of the Project** In my opinion, my project addresses a pivotal challenge in image processing: effectively removing noise to enhance classification accuracy. Image noise, manifesting in forms like Gaussian, impulse, salt, pepper, and speckle, significantly degrades the visual content and clarity. The project's cornerstone is the use of Convolutional Neural Networks (CNN) and deep learning (DL), which have shown exceptional promise in image denoising. Unlike traditional methods like denoising filters, CNN excels in distinguishing between noise, edges, and textures, which is particularly challenging due to their shared high-frequency characteristics. The versatility of CNN is evident in application across various denoising contexts, from supervised to self-supervised denoising, from general image processing to more specialized areas. In addition, a notable aspect of this project is its reliance on MATLAB for implementation. MATLAB stands out in several image and data processing aspects, offering advanced functionality and computational efficiency that often surpasses Python. It hosts numerous features exclusive to its environment, circumventing the compatibility issues often encountered with data types in Python. MATLAB's robust community support further enhances its appeal, providing a rich resource for troubleshooting and knowledge sharing. Importantly, MATLAB has built-in support for various architectures like CNN and UNet, making it an ideal platform for this project. Moreover, the advantages of CNN in this context are manifold. They are adept at removing various types of noise from images, substantially improving denoising capabilities. Several studies have highlighted the superior performance of CNN architectures in image denoising compared to traditional methods. These architectures support end-to-end procedures, allowing for a seamless process from input to denoised output. The implementation of CNN in MATLAB further enhances this efficiency, combining the architectural strengths of CNN with the robust, user-friendly environment of MATLAB. Last but not least, incorporating these elements, the project aims to develop a comprehensive solution for image denoising, leveraging the advanced capabilities of CNN and the robust platform of MATLAB. This combination promises not only to address the current challenges in image denoising but also to set a new standard in the field, offering a powerful tool for researchers and practitioners alike. The outcome will be a software system capable of processing various types of noise, adaptable to different imaging contexts, and providing enhanced accuracy in image classification tasks. In summary, by focusing more on the research, literature review, and implementation aspects, this project aims to make a significant contribution to the field of image denoising and classification, providing valuable insights and tools for further research and practical applications. **Proposed Solution** **Literature Review and Database Selection:** An extensive literature review will be the foundation of this project by searching, reading and summarizing the recent papers. This involves studying various CNN methods used in image denoising, understanding their strengths, weaknesses, and applicability to different noise types. The choice of databases for testing and validation will be crucial. These databases will be selected based on their relevance to real-world scenarios in fields like medical imaging and affective computing.

	Learning and Implementing CNN with MATLAB: The project will leverage MATLAB for implementing CNN models. This involves not only learning the intricacies of MATLAB programming but also understanding and applying various CNN architectures (like DnCNN), and the necessary hardware supports like servers. The focus will be on adjusting parameters specific to the types of noise encountered in the chosen datasets. **Experimentation and Code Debugging & Development:** Experimentation is a key part of this project. This involves setting up the datasets, fine-tuning the CNN models, and implementing the evaluation indices. Debugging and developing robust MATLAB code to handle these tasks is essential. This code development will focus more on the research and implementation aspects as well as creating a software tool. **Expected Outcomes** **Comprehensive Literature Review:** A detailed review of existing literature on CNN-based image denoising techniques. This document will provide insights into various approaches, their effectiveness, and their applicability to different noise types. **MATLAB Implementation:** A suite of MATLAB code implementations for different CNN models tailored for image denoising based on the chosen databases and methods will be finished. This implementation will demonstrate the practical application of theoretical concepts studied in the literature review. **Experimental Reports:** Detailed reports (and PPT) documenting the experimentation process, including the performance of CNN models on different datasets, comparisons with traditional methods, and analyses using various performance metrics will be written. These reports will not only validate the efficacy of CNN in image denoising but also offer insights into the trade-offs between different methods in terms of complexity, speed, and outcomes.
道德规范 **Ethics**	Please confirm by checking the box: ☒ I confirm that I have discussed ethical issues with my supervisor.

Please discuss ethical issues with your supervisor using the ethics checklist in Project Handbook Appendix 1.	Summary of ethical issues: (write “None” if no ethical issues) None
中期目标 **Mid-term target.** **It must be tangible outcomes,** **E.g. software, hardware or simulation.** **It will be assessed at the mid-term oral.**	According to the Gantt Chart, by 4-8 Mar 2024, my project will have achieved key milestones in Task 3: Experimentation, with Task 3.2 and 3.3 in progress. A thorough literature review will guide the selection of classification methods for different noise types in the datasets and methods. These essential for testing our algorithms, will be ready for use. Some strides will be made in developing a preprocessing workflow, setting up the necessary infrastructure for our computational needs. Concurrently, I will have made progress in creating, debugging and refining denoising and classification algorithms, with an initial working code prototype of the denoising software available for initial testing. A comprehensive database of noisy images will be established, serving as the basis for our experimentation. Evaluation indices to assess the quality of image denoising and classification, including accuracy and precision measures, will be implemented. Lastly, a report and presentation summarizing the development process, challenges, and iterative improvements will be prepared for the mid-term oral, offering a clear view of the project's progress and future direction.

Work Plan (Gantt Chart)

Fill in the sub-tasks and insert a letter X in the cells to show the extent of each task

	Nov 1-15	Nov 16-30	Dec 1-15	Dec 16-31	Jan 1-15	Jan 16-31	Feb 1-15	Feb 16-28	Mar 1-15	Mar 16-31	Apr 1-15	Apr 16-30
Task 1 Study and selection of the datasets and classification methods.												
Literature review on image denoising and classification methods	X	X	X	X								
Preliminary study of **MATLAB** and relative deep learning models	X	X	X	X								
Selection of appropriate datasets				X	X							
Study and finalize classification methods				X	X							
Task 2 Design and implementation of the procedures of preprocessing, image data denoising, and classification.												
Design preprocessing workflow					X							
Learn relative software and hardware (E.g. Server)					X	X						
Implement denoising methods					X	X						
Design classification procedures						X	X					
Task 3 Experimentation: definition of the databases; tuning and debugging of the methods; implementation of evaluation indices.												
Database and initial version of code setup						X	X					
Parameters tuning and debugging the codes							X	X	X			
Implement evaluation indices								X	X			
Experimentation									X			
Task 4 Evaluation and reporting of the results.												
Test, analyze and refine according to the results									X	X	X	
Final evaluation											X	X
Prepare final report & supporting documents											X	X
Prepare for final viva slides & final viva												X

Early-term progress report

Include your project early-term progress report here. It must be the final version submitted to QMPlus.

北京邮电大学 本科毕业设计（论文）初期进度报告

Project Early-term Progress Report

<table>
<tr><td>学院
School</td><td>International School</td><td>专业
Programme</td><td colspan="3">Telecommunications Engineering with Management</td></tr>
<tr><td>姓
Family name</td><td>Lin</td><td>名
First Name</td><td colspan="3">Guanyi</td></tr>
<tr><td>BUPT 学号
BUPT number</td><td>2020213069</td><td>QM 学号
QM number</td><td>200977962</td><td>班级
Class</td><td>2020215102</td></tr>
<tr><td>论文题目
Project Title</td><td colspan="5">A Study on Image Denoising Using Deep Learning</td></tr>
</table>

文献综述 **Literature Review**

Abstract

NOISE in images often occurs as isolated pixels or pixel blocks that have a significant visual impact, disrupting the actual information content of the image and making it unclear, which makes it difficult to analyze and negatively impacting subsequent tasks such as image detection, classification, segmentation, tracking and more. So, current image processing often challenges prominently include the effective removal of noise - a disruption that obscures critical information content. This review concentrates on recent advancements in image denoising, emphasizing the application of Deep Learning (DL) methodologies. I explore various noise types (e.g., white Gaussian noise, Poisson noise) and filters (e.g., linear, non-linear, adaptive) used for noise mitigation. Significantly, Convolutional Neural Networks (CNNs) are highlighted for their adaptivity, efficiency, and capability to handle complex noise patterns, outperforming traditional filters. The review categorizes CNN-based denoising strategies for general and specific imagery, detailing their applications, advantages, and the challenges they address. Through this exploration, I underscore the critical role and continual evolution of DL and CNNs in advancing image denoising techniques.

Image Noise & Noise Reduction Filters

Noise can be introduced at different image processing stages when inputs diverge from expectations. One of the great problems in image denoising is to distinguish the noise, edge and texture, which all have high-frequency components. To further discuss, people usually categorize the noise as additive white Gaussian noise (AWGN), impulse noise, quantization noise, Poisson noise, salt-and-pepper noise and speckle noise. Among them, the AWGN often occurs in analog circuitry (E.g. information channels) while the rest of them occur due to faulty manufacturing, bit error, and inadequate photon count.[2]

To mitigate various types of noise, a range of noise reduction filters have been developed, each with its own method of smoothing out disturbances while preserving important image details. These filters fall into six principal categories: linear, non-linear, adaptive, wavelet-based, partial differential equation (PDE), and total variation filters.

Linear filters, which perform noise reduction by correlating output pixels with their neighboring inputs using matrix operations, often blur edges, resulting in a loss of sharpness. On the other hand, non-linear filters like the median filter maintain edge integrity while dampening noise, making them a go-to choice for many applications.

For real-time processing, adaptive filters stand out. They employ statistical methods to dynamically adjust to changing noise patterns, with the least mean square and recursive mean square filters being prime examples of this adaptability.

Wavelet-based filters offer a different approach, transforming images into the wavelet domain to target and reduce additive noise. This method is particularly effective for images with a significant amount of this type of noise.

Despite the effectiveness of these filters in various scenarios, they come with challenges. They may not perform optimally during testing phases, often require manual tuning of parameters, and their reliance on specific denoising models can limit their applicability across different types of noise environments. These limitations highlight the importance of selecting the appropriate filter based on the noise characteristics and the application's requirements. [3]

Filter Type	**Denoising Type**	**Image Quality Comparison**	**Formula and Computational Complexity**	**Additional Notes**
Gaussian Filter	Normal distribution noise (AWGN)	More effective with increasing standard deviation, but causes more blurring	Convolution operation, implemented in time or frequency domain, slower computation	Separable
Mean Filter	Various types of noise	Becomes blurrier with larger kernel sizes	Simple formula, lower computational complexity	Linear filtering method
Median Filter	Especially effective for salt-and-pepper noise	Better edge preservation while denoising	Depends on sorting algorithms, complexity varies with implementation	Non-linear filtering method
Bilateral Filter	Various types of noise except salt-and-pepper noise	Best edge preservation while denoising	Includes spatial and range matrices, more complex computation	Preserves high-frequency information like edges, suitable for complex background images

TABLE 1: Some common filters and their characteristics

Advantages of using CNN in image denoising
To overcome the limitations of traditional noise reduction filters, specifically in image denoising tasks, Convolutional Neural Networks (CNN) is widely used due to the flexibility and advancements:
Adaptivity and Learning: Unlike traditional filters, where parameters are fixed and cannot be adjusted during the filtering process, CNNs can learn and optimize their parameters through network training. This adaptability allows CNNs to perform better in various and changing noise conditions.

Handling Complex Noise Patterns: CNNs are better suited for handling complex noise patterns, which traditional filters might not handle all kinds of them effectively. This is due to the learning capability of CNNs, which can adapt to various noise distributions in the training data.
Preservation of Image Features: CNNs, especially those designed for image denoising, are often more effective in preserving important image features, such as edges and textures, while removing

noise. This is a critical advantage over some traditional filters that might blur or distort these features.

Efficiency in Processing Large Datasets: With the advancement in computational power and the availability of large image datasets, CNNs can be trained more effectively and can process large amounts of data more efficiently than traditional methods.

Versatility and Integration: CNNs can be integrated into broader image processing and computer vision pipelines, offering versatility in applications ranging from basic image enhancement to complex tasks like object detection and scene analysis. [1]

The unique and critical advantage of using CNNs for image denoising lies in their ability to learn from data, adapt to various noise patterns, and preserve important image features while efficiently processing large volumes of data. These aspects make CNNs a forward-looking technology in the field of image processing, with potential for continued improvements and innovations.

Overall, CNN architectures have been enhanced with components such as convolution layers, batch-normalization, ReLU activation functions, and residual learning to improve denoising performance.
Early CNN development faced challenges such as the vanishing gradient problem and hardware limitations, which were overcome by significant advancements like AlexNet in 2012[4], followed by other architectures such as VGG, ResNet[5] and UNet, GoogleNet.

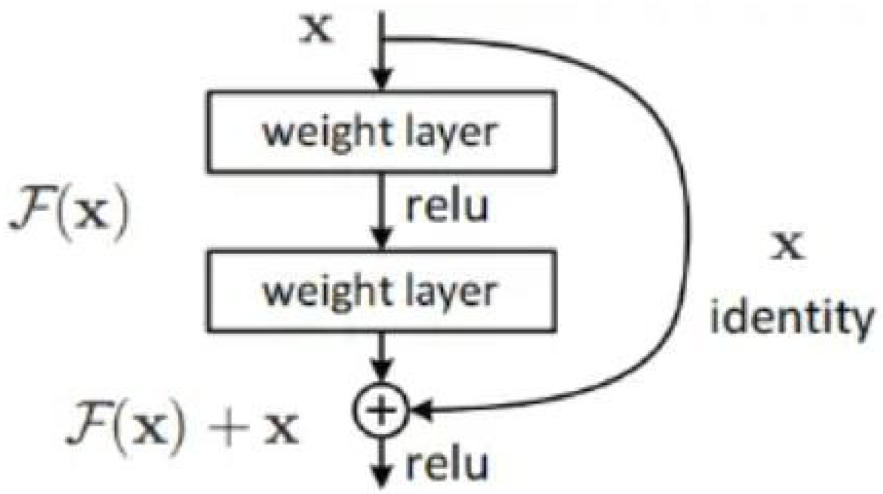


FIGURE 1: A building block of ResNet [5]

Nowadays, more image denoising specified CNNs are introduced to use. For example, Zhang et al. ‘s DnCNN architecture is highlighted for its application in image denoising, super-resolution, and JPEG image deblocking.[6],which can be directly use in MATLAB.This denoising convolutional neural network was trained using manually added AWGN to create noisy-clean image pairs. DnCNN's use of residual learning not only enhanced denoising performance but also significantly reduced computational demands.

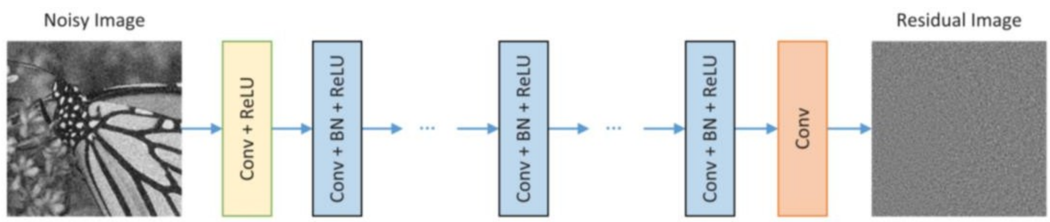


FIGURE 2: The architecture of the DnCNN network[6]

These CNNs have been shown to effectively address drawbacks such as suboptimal test phase performance, manual parameter settings, and model specificity in image denoising comparing with the traditional filters. Also, the capabilities of CNNs extend beyond image denoising to applications in image recognition, robotics, self-driving cars, facial expression recognition, natural language processing, and handwriting digital recognition.[1]

Classification of CNN by application fields
Building upon this versatile foundation, CNNs for image denoising can be systematically categorized into two distinct strategies by application fields:
1. General images (for general proposes), which are utilized for a broad spectrum of purposes. Here, CNN architectures are employed to clean noise from images that are intended for general use.
2. Specific images (for detailed and specialized use) , which are crafted for detailed and specialized applications. In this case, CNNs are tailored to filter noise from images where intricate details are crucial.

These two strategies demonstrate the adaptable nature of CNNs, as they are engineered to handle both the overarching needs of general imagery and the precise requirements of specialized images.

The classification of CNN denoising methods into categories based on image type serves to acquaint the audience with the most current CNN architectures tailored for various image classifications. For a visual representation of these methodologies, refer to the block diagram presented in Figure 1. This delineation sets the stage for a detailed exploration of the distinct approaches applied to both general and specific image denoising, each adapted to its respective image category's requirements.

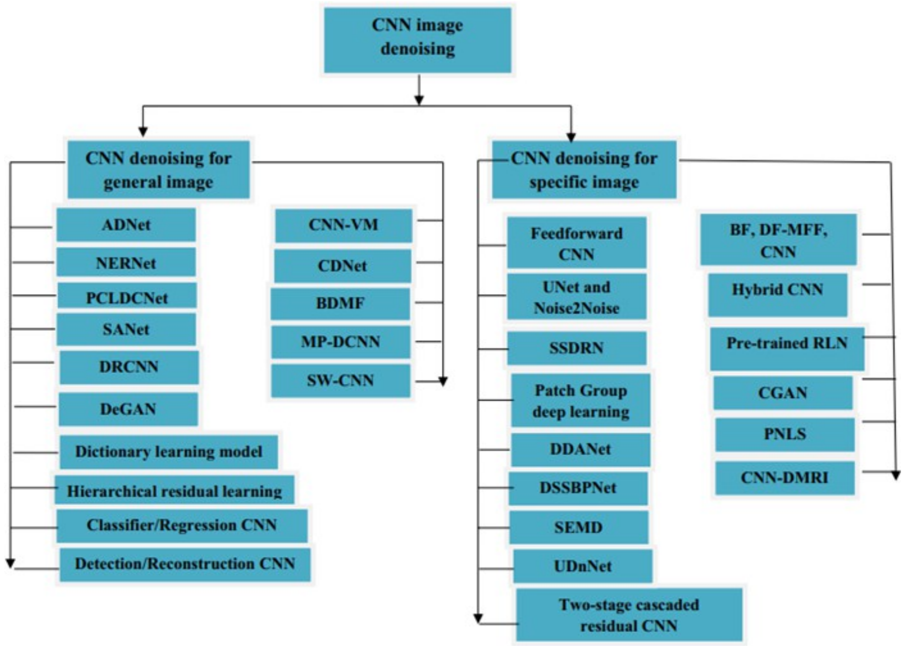


FIGURE 3: CNN image denoising scheme [1]

CNN denoising for general images
There are many CNNs to del with the general images denoising.
Attention-guided denoising CNN(ADNet) is structured into four distinct blocks over 17 layers: the Sparse Block (SB) for enhanced efficiency and reduced depth with 12 layers, the Feature Enhancement Block (FEB) to amplify features through its 4 layers, the single-layer Attention Block (AB) to focus on unknown noise, and the Reconstruction Block (RB) for final image output. Sparsity in SB and attention in AB refine the denoising, while mean square error guides the

training, as depicted in Figure 2.[7]

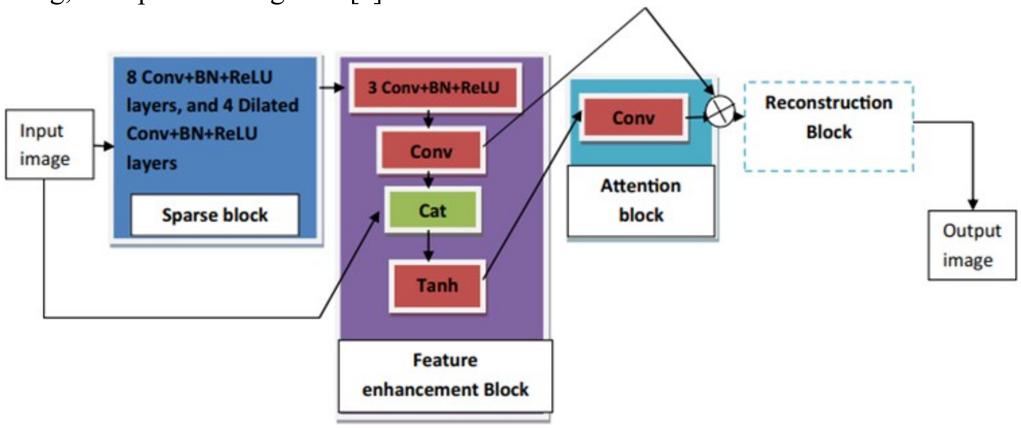


FIGURE 4: Attention-guided denoising CNN [7]

Some CNNs can only get good results of synthetic noise instead of realistic noise. To solve this problem, Noise Estimation and Removal Network (NERNet) was introduced, which consists of two core modules for addressing real noise in images: the noise estimation module utilizes symmetric dilated blocks and pyramid feature fusion to gauge noise levels, while the noise removal module employs this estimation to eliminate noise, merging global and local insights to retain image details and textures. [1]

CNN's proficiency in learning noise patterns and image patches necessitates extensive training data, leading to the creation of the patch complexity local divide and deep conquer network (PCLDCNet). This network segments the learning process into local subtasks, focusing on individual clean image patches, enabling efficient training within their specific local domains.[1]

Another problem of CNN is that the deeper the layer, the higher the error rate (Network Degradation), ResNet is designed to solve it. To make better use of ResNet, Patch Complexity Local Divide and Deep Conquer Network((PCLDCNet) is introduced. The network is partitioned into local subtasks based on clean image patch and conquer block and is trained in its local space. Then each noisy patch weighting mixture is amalgamated with the local subtask, and finally fuse with noise map to estimate.[1]

To further improve, MP-DCNN, an adaptive residual CNN that operates end-to-end. It utilizes leaky ReLU to extract noise and reconstructs image features. Incorporating SegNet, it retrieves edge details from an initial denoised image. The model employs both MSE and a perceptual loss function to produce the final denoised output, as illustrated in Figure 3.[8]

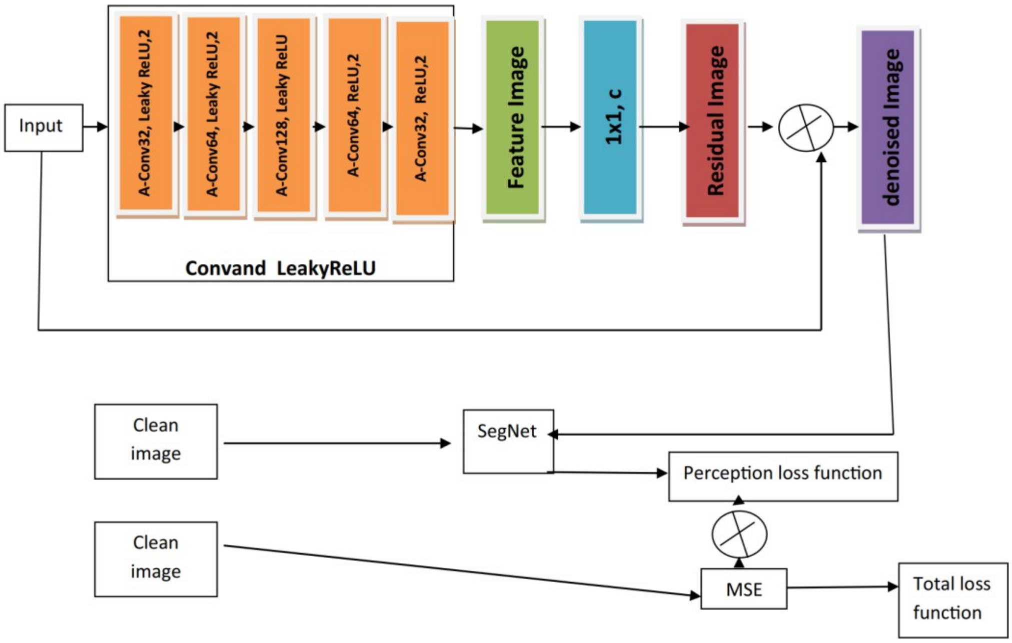


FIGURE 5: MP-DCNN [8]

Furthermore, removing mixed noise from images using CNN is more challenging than addressing single noise types. So DeGAN, a denoising-based generative adversarial network, is introduced to tackle mixed noise. GANs are already established in deep learning, and DeGAN integrates a generator, discriminator, and feature extractor. The generator employs U-Net architecture, while the 10-layer discriminator assesses the noise-free status of images processed by U-Net. The VGG19-based feature extractor aids in training by calculating the loss function, as shown in Figure 4.[9]

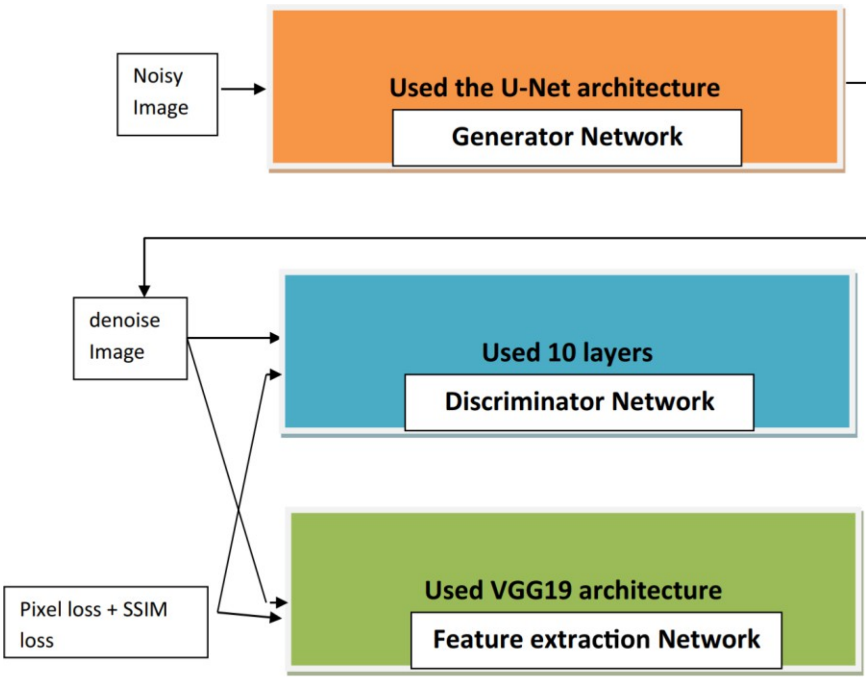


FIGURE 6: DeGAN [9]

To combine classification model and regression model together, a classifier/regression CNN framework for image denoising is introduced. In this system, the classifier network, equipped with convolution, BN, ReLU, softmax, and a skip connection, detects impulse noise. Concurrently, the regression network, comprising four layers and a skip connection, restores noisy pixels identified by the classifier. This restoration is guided by the classifier's predictions, aiming to reconstruct clean images, as detailed in Figure 5.[10]

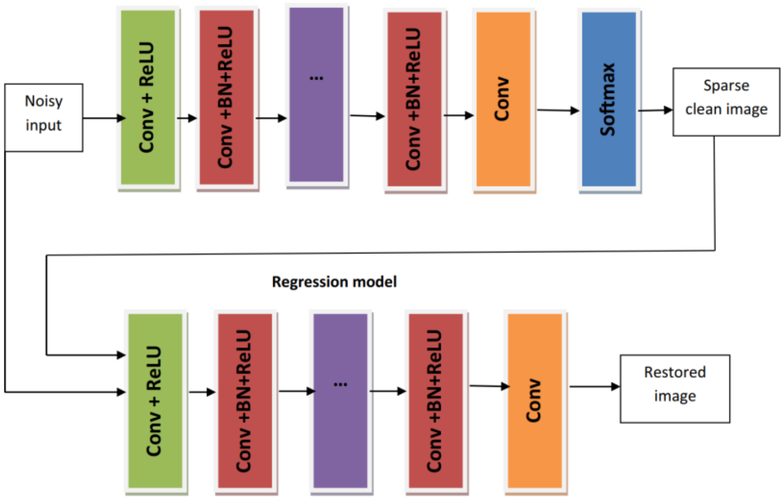


FIGURE 7: Classifer/regression CNN [10]

CDNet is a complex-valued CNN for image denoising. The process begins with feeding the input image into 24 Sequentially Connected Convolutional Units (SCCU), each comprising a complex-valued (CV) convolutional layer, CV ReLU, and CV Batch Normalization (BN). The network utilizes a 64 convolutional kernel, with a residual block employed in the middle 18 units for enhanced performance. To boost computational efficiency, a convolution/deconvolution layer with a stride of 2 is used. The final stage involves a merging layer, which converts complex-valued features back into a real-value image. CDNet is structured into five key blocks: CV Convolution, CV ReLU, CV BN, CV Residual Block (RB), and the merging layer, as depicted in Figure 6. [11]

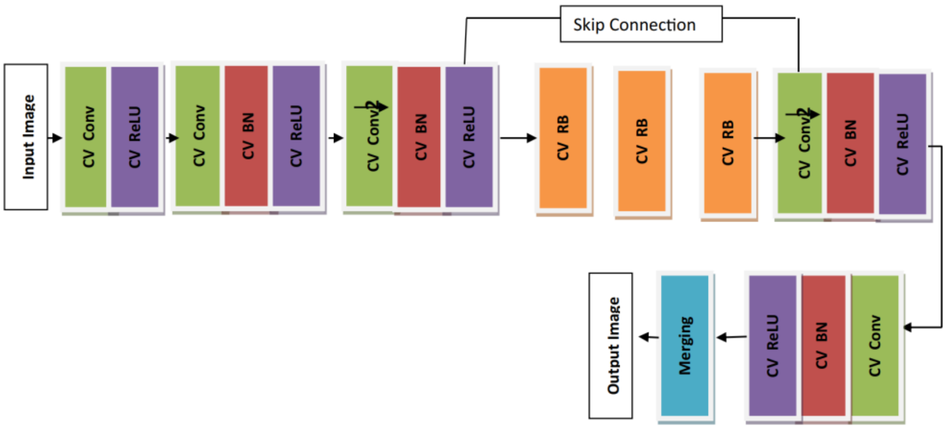


FIGURE 8: Complex value CNN [11]

Method	**Advantages**
ADNet	Mean square error (MSE) refines the training
NERNet	Retains image details and textures when dealing with realistic noise
PCLDCNet	Utilizes ResNet to overcome Network Degradation

MP-DCNN	Incorporating SegNet, it operates end-to-end residual CNN, which extracts noise and reconstructing features of image
DeGAN	Employs GANs, U-Net and VGG-19, which can remove mixed noise
Classifer/regression CNN	Combines classification model and regression model together to reconstruct clean images
CDNet	Complex-valued CNN boosts computational efficiency and outputs real-value image

TABLE 2: Some of CNNs used for general images denoising

In the realm of CNN advancements, several notable implementations and improvements have emerged, each addressing different aspects of image denoising. The Separation Aggregation Network (SANet) utilizes a trio of blocks - the convolutional separation block, deep mapping block, and band aggregation block - to effectively remove noise from images. In contrast, the Detail Retaining CNN (DRCNN) focuses specifically on preserving the integrity of high-frequency image content, ensuring that finer details are not lost during the denoising process. Additionally, the Bayesian Deep Matrix Factorization (BDMF) approach has been designed for multi-image denoising, catering to scenarios where multiple images need simultaneous noise reduction. Another innovative approach, the CNN Variation Model (CNN-VM), also known as EdgeNet, integrates multiple scale residual blocks (MSRB). This design enhances the network's ability to handle image edges and textures, making it particularly effective in retaining crucial image features while removing noise.[1]

CNN denoising for specifc images

There are also various CNNs that can deal with specific images.

Based on previously mentioned DnCNN, the Spectral-Spatial Denoising Residual Network (SSDRN) represents a significant advancement in CNN-based image denoising. This network, notable for preserving the spectral profile while effectively removing noise, operates as an end-to-end algorithm. It incorporates a three-part structure: spectral difference learning, key band selection, and the denoising process, which is executed through the DnCNN model. A distinctive aspect of SSDRN is its utilization of batch normalization layers within each algorithmic block, enhancing its efficiency and effectiveness. Additionally, a novel approach involving patch group deep learning for image denoising was proposed. In this method, a training set is formed from patch groups, which are then processed using advanced deep learning techniques to efficiently reduce noise.[1]

The Two-Stage Cascaded Residual CNN is introduced for the efficient removal of mixed noise from infrared images, integrates a mixed convolutional layer. This layer combines various convolution types, such as dilated, sub-pixel, and standard convolutions, to enhance feature extraction and accuracy. The model employs residual learning to estimate calibration parameters from the input image accurately. It features Five Feature Extraction Blocks (FEBs) that utilize the Coarse–Fine Convolution Unit (CF-Conv) and the Spatial–Channel Noise Attention Unit (SCNAU), effectively stacking noise features. Each network's final convolution layer is equipped with a uniquely designed single filter, optimized for noise reduction in infrared images. [12]

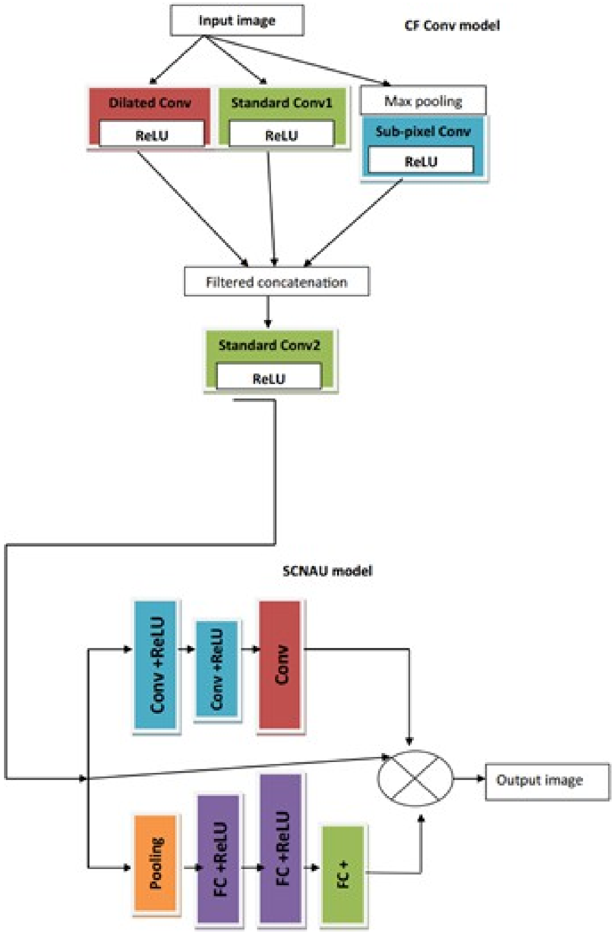


FIGURE 9: Two-phased cascaded residual CNN [12]

To get better use of ResNet and pretraining methods to deal with specific images, a novel approach for despeckling ultrasound images using a pre-trained Residual Learning Network (RLN) is introduced. This model comprises both a noise model, created from a training dataset, and the pre-trained RLN itself. The process involves generating random patches from speckle noise images and using the RLN to train these patches. The RLN, featuring 59 layers that include convolutional layers, ReLU, and batch normalization, is then employed to produce despeckled images. This method was rigorously tested on both artificial and naturally speckle noise-corrupted images, demonstrating its effectiveness in ultrasound image enhancement. [13]

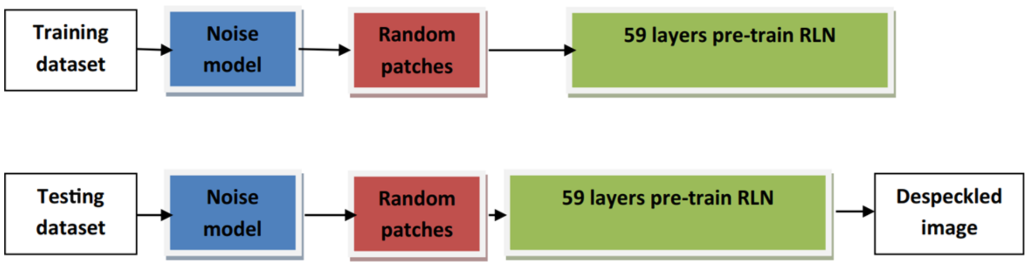


FIGURE 10: Pre-trained RLN [13]

Building upon the previous advancements in image denoising, Progressive Network Learning Strategy (PNLS) tailors for images following the Rician distribution. This approach utilizes large convolutional filters and is structured around two distinct residual blocks. The first of these blocks is designed for fitting the pixel domain, comprising convolution and ReLU layers, notably omitting the batch normalization (BN) layer. In contrast, the second block, aimed at matching pixel domains, includes convolution, ReLU, and BN layers. Each block is composed of 5 layers, with three convolution layers positioned strategically between the two blocks to facilitate efficient processing and enhance denoising performance. [14]

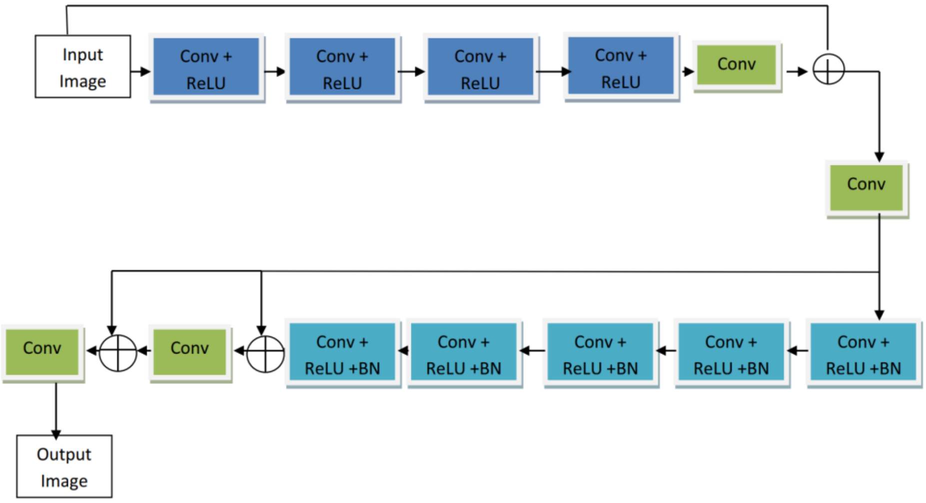


FIGURE 11: PNLS [14]

Methods	**Advantages**	**Application fields**
SSDRN	Executes through the DnCNN, using patch groups to form training sets	Spectral profile preservation
Two-phased cascaded residual CNN	Employs residual learning to estimate calibration parameters, uniquely designed single filter	Infrared images denoising
Pre-trained RLN	Comprises both a noise model, created from a training dataset, and the pre-trained RLN itself.	Ultrasound image enhancement
PNLS	Utilizes large convolutional filters and two distinct residual blocks to facilitate efficient processing and enhance denoising performance	Images following the Rician distribution

TABLE 3: Some of CNNs used for specific images denoising

In the realm of specialized image denoising, several innovative methods have been developed for targeted applications. UDnNet, a generative adversarial network, is specifically crafted for denoising underwater images. This network is composed of two main components: a generator network, which focuses on generating clear images from noisy underwater inputs, and a discriminator network that evaluates the authenticity of the generated images. Additionally, the Hybrid CNN Method addresses speckle noise reduction, employing a unique strategy. This method starts with networks trained on Gaussian noise models and then fine-tunes them using data that emphasize structural boundaries, effectively combining general noise reduction principles with specific structural considerations. Moreover, the CNN-DMRI represents a breakthrough in MRI scan denoising. This method is built around an encoder-decoder structure, meticulously designed to retain essential image features while efficiently filtering out irrelevant noise components. This tailored approach ensures the preservation of critical details in MRI scans, enhancing the clarity and usability of the resulting images.[1]

In the field of image denoising, deep learning approaches can also be divided into supervised or self-supervised categories. This classification is based on whether the methods require pairs of noisy and clean images for training.

Supervised image denoising
Supervised image denoising is a process where a deep neural network (DNN) is trained using pairs of noisy and corresponding clean images. In this approach, the model is taught to map a noisy image to its clean counterpart. This method can be implemented in two main ways: either by directly learning the transformation from a noisy image to a clean image, as cited in several studies, or by targeting the residual difference between a clean and a noisy image, which effectively teaches the model to separate noise from the actual image content.[15]
Once the model is adequately trained, it becomes capable of processing new, unseen noisy images and producing their denoised versions. This capability is particularly valuable in various applications such as medical imaging, astronomical observation, and photography, where maintaining the integrity of the original image is crucial.

Jain et al. [16] pioneered the use of DNNs for image denoising in 2008, achieving results on par with then-leading traditional algorithms like wavelet and MRF methods, but with reduced computational costs. This breakthrough led to the development of more DNN-based methods.

Various approaches have been developed to handle different types of noise. Alongside the common Additive White Gaussian Noise (AWGN), noise models like Poisson-Gaussian distribution are also utilized.

In a separate development, Guo et al. [17] proposed the CBDNet, specifically tailored for real-world photography. This network, embracing the Poisson-Gaussian noise model and in-camera processing, consists of two parts: a noise estimation subnetwork and a non-blind denoising subnetwork. The noise estimation subnetwork employs an asymmetric loss function, penalizing underestimation of noise more heavily, thereby improving the robustness of the denoiser.

Following the framework established by CBDNet, a simpler yet effective model called the SDNet was introduced [18]. It utilizes the generalized signal-dependent noise model and achieves competitive results on both synthetic and real noisy images through a stage-wise process and lifted residual learning. Each of these developments represents a stride forward in the field, offering unique solutions to the complex problem of image denoising.

Besides DNNs, the UNet model, initially established for biomedical image segmentation, has also evolved significantly, finding applications in various domains including image denoising. Its lightweight and high-performance characteristics make it an ideal baseline model for semantic segmentation tasks. The core design of UNet involves the use of convolutional neural networks (CNNs) which have shown excellent performance in medical image segmentation. One key aspect of UNet's training involves the use of stochastic gradient descent and data augmentation techniques to enhance its invariance and robustness properties, particularly in scenarios with limited training samples. This is crucial for its application in medical imaging, where acquiring large annotated datasets can be challenging.

The Dense U-Net (DDUNet)[20] represents an improved version of the original UNet, focusing on applications in image denoising and segmentation. It capitalizes on the strengths of UNet's architecture while enhancing it with additional features to better handle the complexities involved in these tasks.

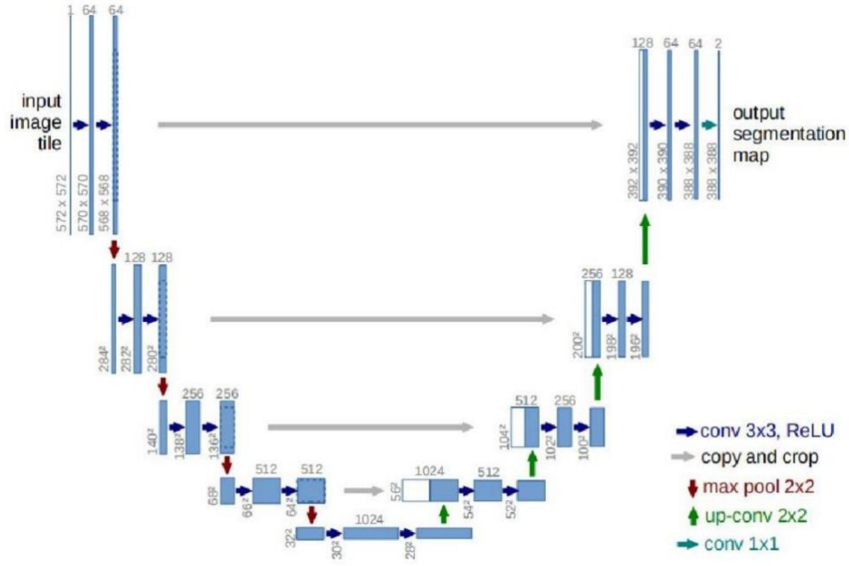


FIGURE 12: the network architecture of UNet [20]

Similarly, the Residual Dense U-Net (RDUNet)[21] introduces densely connected convolutional layers within its encoding and decoding segments. This adaptation aims to improve the denoising performance of the network, leveraging the dense connections to capture more intricate image details.

In a shift towards transformer-based models, the Swin Transformer UNet (SUNet)[22] represents a significant departure from traditional CNNs. It combines the strengths of both transformers and CNNs, utilizing shifted windows to reduce computational complexity while maintaining high performance in image restoration tasks.

Lastly, the Multi-Task Attentional U-Net (MTA-Net)[23], designed for Hyperspectral image (HSI) denoising, introduces a unique approach. It features specialized networks and learning strategies that cater to the specific challenges posed by HSI data, including the need for precise noise estimation and effective noise separation.

Each of these UNet variants demonstrates the model's adaptability and effectiveness across different domains, particularly in image denoising and segmentation tasks, highlighting its ongoing evolution and significance in the field of image processing.

Supervised image denoising effectiveness hinges on diverse and quality training data, but acquiring perfect noisy-clean image pairs is challenging. This has led to a shift towards self-supervised methods, which are more flexible as they don't always require clean images. Enhancements in CNNs and GANs further improve these models, allowing for better texture and structure recreation in images. However, the dependency on extensive training data and the practical difficulty of obtaining perfectly clean images in real-world scenarios are significant hurdles. Consequently, researchers often use artificial noise addition to clean datasets, creating synthesized pairs for training, though this doesn't always accurately mimic real-world noise.

Different from supervised algorithm, Self-supervised image denoising algorithm is a deep learning image denoising algorithm that does not require paired noisyclean images as training data.

BSN (Blind Spot Network), a self-supervised image denoising algorithm, operates on the principle that noise is spatially independent and has a zero mean. It leverages the spatial correlation of image signals to predict blind pixels using surrounding pixels. Recently, numerous BSN-based image denoising algorithms have been developed, indicating the method's growing significance in the field.

Additionally, the Transformer, a model known for its ability to extract global information in image processing tasks, has shown to have unique advantages over CNNs. This has led to the categorization of self-supervised image denoising algorithms into three types: General methods, BSN-based methods, and Transformer-based methods. Each type offers distinct approaches and benefits, contributing to the diversity and effectiveness of image denoising techniques in contemporary research.

General self-supervised image denoising methods

The realm of image denoising has witnessed significant advancements through the adoption of self-supervised learning techniques. These methods have diversified the strategies used for reducing noise in images, each with its unique approach and application. This overview presents a variety of general self-supervised image denoising methods, highlighting their respective applications, advantages, and other relevant notes. From methods like Noise2Noise (N2N) to more sophisticated approaches like CVF-SID, the landscape of image denoising is rich with diverse solutions tailored to different noise types and scenarios.

Method	Denoising Application	Advantages	Additional Notes
N2N	Gaussian, Poisson, Bernoulli noise denoising and random text overlays remove.	Precisely aligned noisy-noisy image pairs.	Paired noise images are needed
GCBD	Real-world sRGB image noise, Gaussian and Mixture noise denoising.	GAN-generated noise distribution applied to clean images for synthetic noisy-clean pair creation.	Unpaired clean images
SURE-based Method	Gaussian noise	Stein's unbiased risk estimator(SURE) based method for refined risk estimation.	Noise Model
Noisier2noise	Gaussian additive noise and multiplicative Bernoulli noise denoising.	Synthetic noise addition to original noisy images for label generation, followed by applying similar noise types to these labels for input creation.	Arbitrary noise model
Recorrupted to-recorrupted (R2R)	AWGN and real-world sRGB image noise denoising.	Data augmentation technique for generating noisy-noisy pairs through re-corruption.	Noisy level function(NLF) or ISP function
NBR2NBR	Gaussian, Poisson noise and real-world rawRGB image noise denoising.	Creating noisy-noisy pairs by splitting a single noisy image into two sub-noise images.	
Noise2Score	Gaussian, Poisson, and Gamma noise denoising.	Training a Neural Network to estimate the score function and using Tweedie's formula for final denoising is an effective method for handling various exponential family noises.	Arbitrary noise model

NAC	AWGN and real-world sRGB image noise denoising.	Introducing synthetic noise to images with existing weak noise for inputs, using the original weak noise images as targets.	Noise model
CVF-SID	Real-world sRGB image noise denoising.	Cyclic multi-Variate Function (CVF) employs a CNN model to split sRGB noise images into clean, signal-independent, and signal-dependent noise components.	
IDR	Gaussian, binomial and impulse noise, real-world raw noise denoising.	Enhancing denoising performance using iterative techniques in the model.	Noise model

TABLE 4: Some General self-supervised image denoising methods

The field of self-supervised image denoising continues to evolve, offering a range of solutions for various types of noise in different imaging contexts. Each method presents a unique set of advantages, whether it's the precision of noisy pair alignment, the generation of synthetic noisy-clean pairs, or the employment of advanced estimators and functions. The diversity of these methods underscores the dynamic nature of image denoising research and its ongoing pursuit to develop more efficient, accurate, and versatile denoising techniques suitable for the challenges of real-world image processing.

BSN-based self-supervised image denoising methods

BSN-based self-supervised image denoising is an innovative method that enhances image quality by predicting noise-free pixels of masked pixels. This technique relies on the spatial continuity between masked pixels and their surrounding counterparts in the image signal. BSN's effectiveness hinges on the premise that image noise is spatially independent and has a zero-mean, whereas the image signal itself demonstrates spatial correlation.

BSN-based methods are primarily categorized into two distinct strategies based on their approach to masking: 'mask in input' and 'mask in network'.

The 'mask in input' approach, as seen in methods like N2V (Noise2Void) , N2S (Noise2Self) , PN2V (Probabilistic Noise2Void) , Noise2Same , S2S (Self2Self) , B2UB (Blind2Unblind) , and others, involves masking certain pixels in the noisy image. This masked image serves as input, with the complete noisy image used as the target for supervised training on deep neural networks.

Method	Noise Type	Applications	Advantages
N2V (Noise2Void)	Gaussian and biomedical	Biomedical imaging	Independently masks random pixels, suitable for pixel-wise noise.
N2S (Noise2Self)	Gaussian (blind)	General	Utilizes J-invariant function for masking, introducing randomness effectively.

PN2V (Probabilistic Noise2Void)	Arbitrary noise models	Microscopy, low-light imaging	Employs probabilistic modeling for accurate intensity prediction.
Noise2Same	Gaussian	General	Adopts J-invariant masking, replaces pixels with local averages for consistency.
S2S (Self2Self)	Gaussian, salt-and-pepper, sRGB	General	Bernoulli sampling creates effective noisy pairs for diverse noise types.
B2UB (Blind2Unblind)	FMDD, Gaussian, Poisson, rawRGB	General	Features global-aware masking and re-visible loss for enhanced denoising.

TABLE 5: BSN-based self-supervised image denoising methods of 'mask in input'

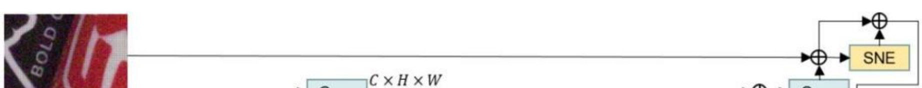
Conversely, the 'mask in network' strategy, exemplified by methods such as Laine et al. , DBSN , AP-BSN , MM-BSN , and Li et al. , focuses on masking parts of the receptive field during feature extraction within the network structure. This approach enables the model to use surrounding pixels of the feature maps to predict the target pixel.

Method	**Noise Type**	**Advantages**
Laine et al.	Gaussian, Poisson, Impulse	Utilizes masking in four directions to create a blind spot network, enhancing denoising capability.
DBSN	AWGN, HG, MG, real-world sRGB, Unpaired clean images	Incorporates dilated convolution, NLF, and knowledge distillation, effective for various noise models.
AP-BSN	Real-world sRGB	Features asymmetric P D in training/testing and a random-replacing refinement post-process.
MM-BSN	Real-world sRGB	Implements a multi-mask strategy for large area noise, enhancing spatial noise reduction.
Li et al.	Real-world sRGB	Differentiates between flat and textured regions, creating tailored supervisions for each.

TABLE 6: BSN-based self-supervised image denoising methods of 'mask in network'

Self-supervised image denoising based on Transformer

Transformers, originally excelling in natural language processing, have also made significant strides in the field of computer vision, including image denoising. However, applying a pure Transformer model directly to self-supervised image denoising can result in sub-optimal outcomes. To address this, the Context-aware Denoise Transformer (CADT)[24] was developed, enhancing the synergy between Transformer and CNN technologies.It employs a dual-branch structure, combining global and local features. The global branch uses a Transformer encoder, specifically a window-based multi-head Transformer encoder similar to the one in Swin Transformer, to capture global image information. This encoder includes a multi-head self-attention (MSA) module and a multi-layer perception (MLP) module. In contrast, the local branch focuses on extracting local contextual information using several convolution layers, including deformable convolutions. This approach allows CADT to effectively retain important image details during the denoising process. Figure 2 presents the architecture of Denoise Transformer.



Method	**Mask Way**	**Applications**	**Advantages**
DT	Mask in inputs	FM dataset, Gaussian, Poisson, and real-world rawRGB images noise denoising	Combines CNN and Transformer for enhanced denoising.
LG-BPN	Mask in networks	Real-world sRGB image noise denoising	Uses DSPMC for blind spot creation and noise break; DTB for integrating local and global information.
SwinIA	Mask in inputs	FM dataset, Gaussian, Poisson, and real-world rawRGB images noise denoising	Employs Transformer for efficient image denoising.

TABLE 7: Some other transformer-based self-supervised image denoising methods.

These developments highlight the ongoing evolution in the field of image denoising, where traditional CNN approaches are being augmented or even replaced by more advanced Transformer-based models, offering improved performance in terms of both efficiency and effectiveness.

Application-Specific Denoising:
In the domain of application-specific image denoising, different techniques cater to varied and specialized needs across fields. Image denoising is pivotal in enhancing the quality and usability of images in diverse scenarios.

For instance, in medical imaging, especially for detecting diseases such as COVID-19, CT image denoising plays a critical role. Advanced deep learning networks are employed to filter out noise from CT scans, thereby improving the clarity and accuracy of disease detection. This process is crucial for timely and precise diagnosis, which directly impacts patient outcomes.[26]

Similarly, in neuroscience, the technique of "Fast, efficient, and accurate neuro-imaging denoising via supervised deep-denoising method" finds application in various settings such as whole-brain imaging, large-field-of-view imaging, and the detailed analysis of complex neurite structures. These applications demand high precision and clarity, given the intricate nature of neuroimaging data.[27]

Moving forward, Prior Residual Noise Embedded Denoising Diffusion Probabilistic Models (Resfusion) [28] is a cutting-edge approach that integrates end-to-end models with denoising diffusion models, consisting of a training pipeline and an inference pipeline. It's designed specifically for image segmentation tasks, utilizing the gradual image generation capabilities of diffusion models. This innovative method streamlines the process of image denoising, restoration, and even complex tasks like image dehazing and deraining, marking a significant advancement in the field.

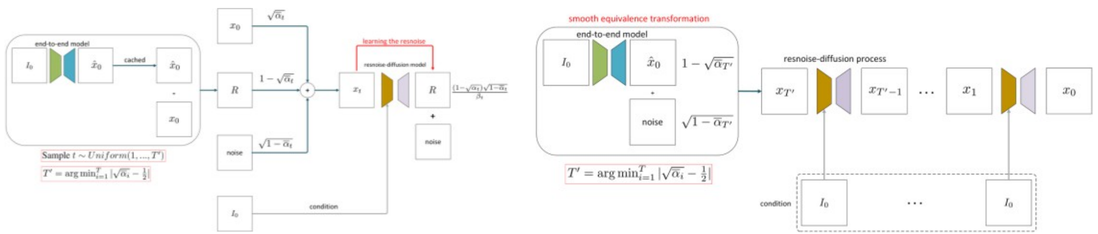


FIGURE 15: The training pipeline and inference pipeline of Resfusion [28]

On the other hand, Unsupervised Image Denoising via Self-Collaboration Parallel Generative Adversarial Branches (SCPGabNet)[29] offers a solution to the challenge of training without paired datasets. SCPGabNet uses a self-collaboration strategy, allowing the network to self-improve without increasing complexity or altering its architecture, showing better results compared to some other denoising methods. This technique is especially beneficial in scenarios where acquiring paired training images is difficult, demonstrating the adaptability of denoising methods to practical, real-world challenges.

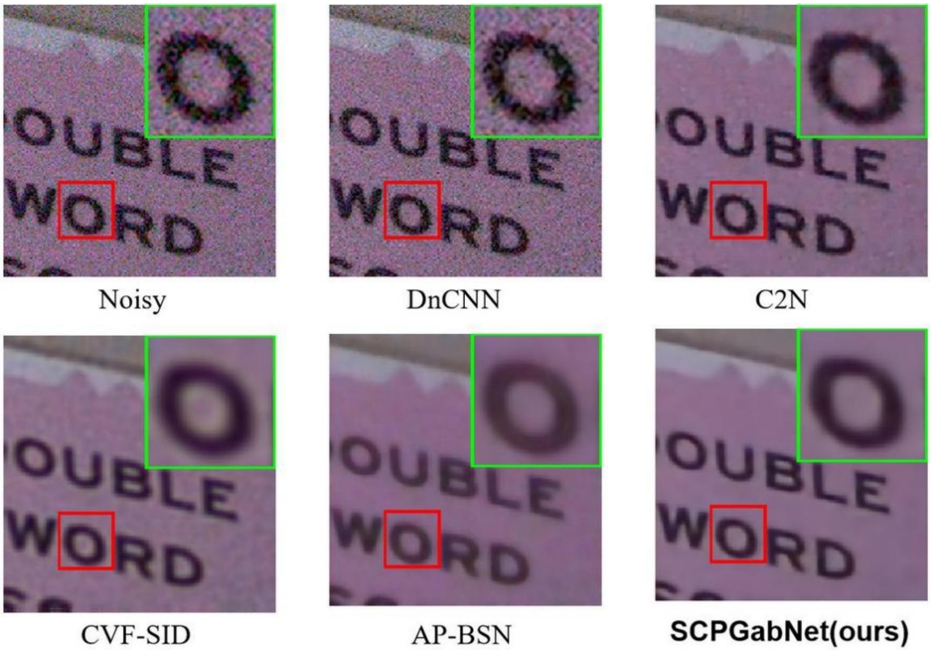


FIGURE 16: A real noisy image from the SSID Validation dataset and results from different methods.[29]

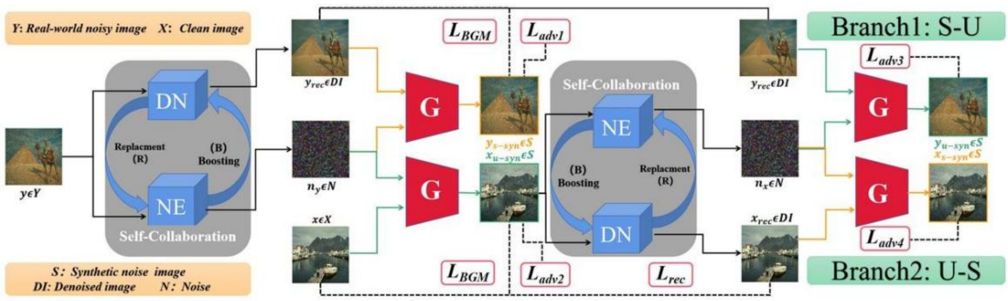


FIGURE 17: The architecture of SCPGabNet framework[29]

Detail Reconstruction in CNN-based Image Denoising Algorithms: Despite advancements, CNN-based image denoising algorithms still face challenges in detail reconstruction, often resulting in over-smoothed images. New approaches like diffusion models and LSTM-enhanced DnCNNs are being explored to preserve high-frequency details and address these shortcomings.[30]

End-to-End Fully Unsupervised Denoising Approaches: Fully unsupervised denoising, crucial for scenarios lacking paired training datasets, is gaining traction. Innovative models using GANs and VAEs, like unified end-to-end deep learning models and frequency-sensitive methods, are showing promise in addressing diverse noise types without relying on paired datasets.[31]

These studies demonstrate the ongoing evolution in the field of image denoising, leveraging advanced machine learning techniques to address specific challenges in various applications. The integration of diffusion models and generative adversarial networks (GANs) in these methods highlights the trend towards more sophisticated, efficient, and versatile denoising techniques.

Conclusion

This review encapsulates some of the recent developments in image denoising, showcasing the paradigm shift towards Deep Learning, particularly Convolutional Neural Networks (CNNs). I have discussed various filters and CNN architectures, highlighting their specific applications in noise reduction across diverse imaging contexts. The adaptability of CNNs to different noise patterns and their proficiency in preserving essential image features while processing large datasets efficiently marks a significant advancement in image processing. The exploration of application-specific denoising techniques underscores the versatility and potential of DL in addressing complex real-world challenges in image denoising. This review, while focusing on some of the recent progress, also opens avenues for my research, particularly in the realms of supervised, self-supervised, and transformer-based denoising methods.

Reference List

[1] Methods for image denoising using convolutional neural network:
a review
Ademola E. Ilesanmi1,2 · Taiwo O. Ilesanmi

[2] Goyal B, Dogra A, Agrawal S, Sohi BS, Sharma A (2020) Image denoising review: from classical to state-of-the-art approaches. Inform Fusion 55:220–244

[3] Fan L, Zhang F, Fan H et al (2019) Brief review of image denoising techniques. Vis Comput Ind Biomed Art 2:7. https:// doi.org/10.1186/s42492-019-0016-7

[4] Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classifcation with deep convolutional neural networks. In: Advances
in Neural information Processing Systems, pp 1097–1105

[5] Deep Residual Learning for Image Recognition
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun(2015)

[6] Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Trans Image Process 26(7):3142–3155

[7] Schwenker F, Kestler HA, Palm G (2001) Three learning phases for radial-basis-function networks. Neural Netw 14(4):439–458

[8] Gai S, Bao Z (2019) New image denoising algorithm via improved deep convolutional neural network with perceptive loss. Expert Syst Appl 138:112815

[9] Lyu Q, Guo M, Pei Z (2020) DeGAN: mixed noise removal via generative adversarial networks. Appl Soft Comp J 95:106478

[10] Jin L, Zhang W, Ma G, Song E (2019) Learning deep CNNs for impulse noise removal in images. J Vis Commun Image R 62:193–205

[11] Quan Y, Chen Y, Shao Y, Teng H, Xu Y, Ji H (2021) Image denoising using complex-valued deep CNN. Pattern Recogn 111:107639

[12] Guan J, Lai R, Xiong A, Liu Z, Gu L (2020) Fixed pattern noise reduction for infrared images based on cascade residual attention CNN. Neurocomputing 377:301–313

[13] Kokil P, Sudharson S (2020) Despeckling of clinical ultrasound images using deep residual learning. Comp Methods Programs Biomed 194:105477

[14] Li S, Zhou J, Liang D, Liu Q (2020) MRI denoising using progressively distribution-based neural network. Magn Reson Imaging 71:55–68

[15] D. Zhang, F. Zhou, X. Yang, and Y. Gu, "Unleashing the Power of Self-Supervised Image Denoising: A Comprehensive Review," arXiv:2308.00247v3 [eess.IV], Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.00247.

[16] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. Advances in Neural Information Processing Systems, 21, 2008.

[17] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In Pro ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1712–1722, 2019

[18] Hengyuan Zhao, Wenze Shao, Bingkun Bao, and Haibo Li. A simple and robust deep convolutional approach to blind image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pages 0–0, 2019

[19] M. Elad, B. Kawar, and G. Vaksman, "Image Denoising: The Deep Learning Revolution and Beyond—A Survey Paper," 2023. [Online]. Available: https://doi.org/10.1137/23M1545859.

[20] J. Cheng, S. Tian, L. Yu, S. Liu, C. Wang, Y. Ren, H. Lu, and M. Zhu, "DDU-Net: A dual dense U-structure network for medical image segmentation," Applied Soft Computing ,2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1568494622004860?dgcid=coauthor.

[21] J. Gurrola-Ramos, O. Dalmau, and T. E. Alarcón, "A Residual Dense U-Net Neural Network for Image Denoising," IEEE, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9360532.

[22] C.-M. Fan, T.-J. Liu, and K.-H. Liu, "SUNet: Swin Transformer UNet for Image Denoising," 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9937486/authors#authors.

[23] F. Xiong, Z. Gu, W. Zheng, T. Li, and J. Zhou, "Multi-Task Attentional U-Net for Hyperspectral Image Denoising," School of Computer Science and Engineering, Nanjing University of Science and Technology, China, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10283365.

[24] D. Zhang and F. Zhou, "Self-Supervised Image Denoising for Real-World Images with Context-aware Transformer," 2023.

[25] C. Tian, M. Zheng, W. Zuo, S. Zhang, Y. Zhang, and C.-W. Ling, "A cross Transformer for image denoising," 2023. [Online]. Available: https://arxiv.org/abs/2310.10408.

[26] S. U. Khan, I. Ullah, N. Ullah, S. Shah, M. El Affendi, and B. Lee, "A novel CT image de-noising and fusion based deep learning network to screen for disease (COVID-19)," Nature, 2023. [Online]. Available: https://www.nature.com/articles/s41598-023-33614-0#:~:text=,Nature%E2%80%A0www.nature.com%E3%80%91.

[27] S. Chaudhary, S. Moon, and H. Lu, "Fast, efficient, and accurate neuro-imaging denoising via supervised deep learning," Nature, 2023. [Online]. Available: https://www.nature.com/articles/s41467-022-32886-w#article-comments.

[28] S. Zhenning, D. Changsheng, P. Bin, X. Xueshuo, H. Along, Q. Qiaoying, and L. Tao, "Resfusion: Prior Residual Noise embedded Denoising Diffusion Probabilistic Models," 2023. [Online]. Available: https://arxiv.org/abs/2311.14900. [Accessed: 25 Nov 2023].

[29] X. Lin, C. Ren, X. Liu, J. Huang, and Y. Lei, "Unsupervised Image Denoising in Real-World Scenarios via Self-Collaboration Parallel Generative Adversarial Branches," 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2308.06776. [Accessed: 13 Aug 2023].

[30] J. Zhang, J. Zhang, and H. Hong, "Image denoising based on improved RDN algorithm," IEEE, January 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9994352#:~:text=,January%202023%20ISBN%20Information.

[31] V. C. Dodda, L. Kuruguntla, K. Elumalai, S. Chinnadurai, J. T. Sheridan, and I. Muniraj, "A denoising framework for 3D and 2D imaging techniques based on photon detection statistics," Nature, 2023. [Online]. Available: https://www.nature.com/articles/s41598-023-27852-5.

是否符合进度？**On schedule as per GANTT chart?**
YES

下一步 **Next steps:**
1. Continue studying the classification methods and selecting the data sets
2. Go to UPV for further study and work
3. Design preprocessing workflow and implement the denoising methods

Mid-term progress report

Include your project mid-term progress report here. It must be the final version submitted to QMPlus.

北京邮电大学 本科毕业设计（论文）中期进度报告

Project Mid-term Progress Report

学院 **School**	International School	**专业** **Programme**	**Telecommunications Engineering with Management**		
姓 **Family name**	Lin	**名** **First Name**	Guanyi		
BUPT 学号 **BUPT number**	2020213069	**QM 学号** **QM number**	200977962	**班级** **Class**	2020215102
论文题目 **Project Title**	A Study on Image Denoising Using Deep Learning				

是否完成任务书中所定的中期目标？Targets met (as set in the Specification)?
YES

已完成工作 Finished work:

Comprehensive Literature Review

Following my tutor's suggestions and the project specification, I have written a detailed literature review, which encapsulates some of the recent developments in image denoising, showcasing the paradigm shift towards Deep Learning, particularly Convolutional Neural Networks (CNNs). I have discussed various filters and CNN architectures, highlighting their specific applications in noise reduction across diverse imaging contexts. To gain a more in-depth understanding of Image Denoising Using Deep Learning, I meticulously read and summarized the literature recommended by my mentor. Additionally, I conducted searches for the latest research developments in this field through SCI, Nature, IEEE, and Arxiv, and compiled summaries of these essays.

This comprehensive review not only highlights several significant advancements in the field but also boosts my own research work. It casts light on a variety of innovative approaches, especially within the domains of supervised learning, which is used to deal with both general images and the specific fields; self-supervised learning with various algorithms and networks; and also the cutting-edge transformer-based denoising methods, which utilize attention mechanisms to model global dependencies within the data. These areas present a lot of opportunities for exploration and development, giving good ways to make noise reduction algorithms better.

In summary, the literature review I have compiled is not just a mere recount of existing knowledge but a carefully constructed document that highlights the past developments while showing the way for cutting-edge research. This detailed overview provides a basic knowledge of the present difficulties and opportunities for achieving clear image quality despite noise.

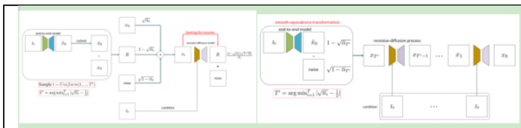
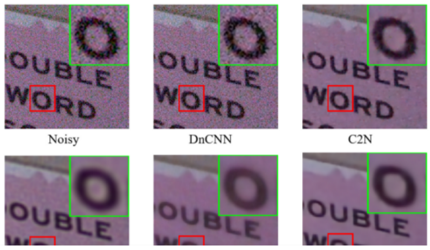


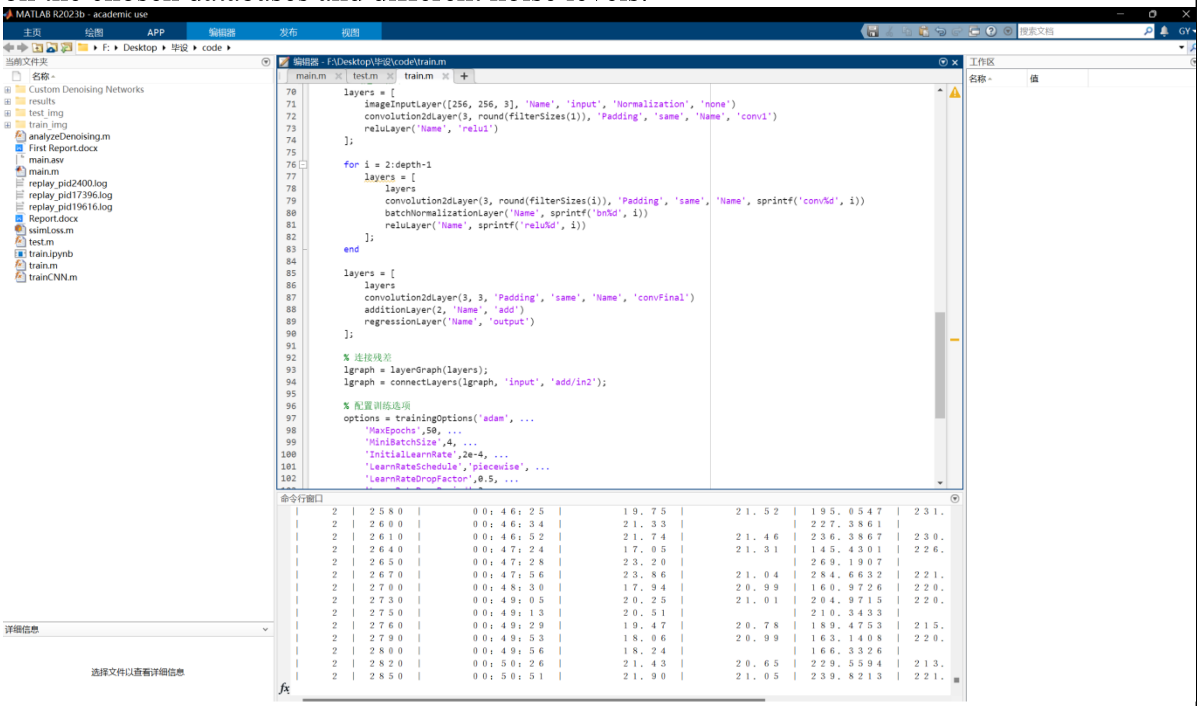
FIGURE 15: The training pipeline and inference pipeline of Resfusion [28]

On the other hand, Unsupervised Image Denoising via Self-Collaboration Parallel Generative Adversarial Branches (SCPGabNet)[29] offers a solution to the challenge of training without paired datasets. SCPGabNet uses a self-collaboration strategy, allowing the network to self-improve without increasing complexity or altering its architecture, showing better results compared to some other denoising methods. This technique is especially beneficial in scenarios where acquiring paired training images is difficult, demonstrating the adaptability of denoising methods to practical, real-world challenges.



MATLAB Implementation & Result Analysis

In addition, after studying how to implement CNN in MATLAB, I have also implemented the MATLAB code for different CNN models that are tailored for image denoising based on the chosen databases and different noise levels:



The code is structured to handle various noise types and trains my own Convolutional Neural Networks (CNN) for the denoising task. Below is the overview of the implementation process:

Image Preprocessing

The code begins by gathering images (dataset) from a specified directory, supporting various formats (PNG, JPG, JPEG, and BMP)

Each image is resized to 256x256 pixels to maintain consistency.

Grayscale images are converted to RGB by replicating the single channel across the three color channels.

The pixel values of the images are converted to double precision for further processing.

Noise Addition
Depending on the noiseType parameter (gaussian, salt & pepper, speckle, Poisson), the code adds corresponding noise to the images.
This process generates a set of noisy images alongside the original clean images, which are crucial for the training of the denoising network.
For each image, the variance of the noise added is a randomly generated value, ranging from 0.001 to 0.02, to ensure diversity in the noise levels processed by the CNN during the training process.

Dataset Splitting
The training dataset is randomly split into a training set (90%) and a validation set (10%).
This split ensures that the network can be trained on a large dataset while being validated on unseen data to check for overfitting.

Neural Network Architecture
The CNN starts with an image input layer tailored to the size of the processed images.
It consists of a series of convolutional layers, each followed by batch normalization and ReLU activation.
The depth of the network is dynamically set to 10 layers, with the number of filters increasing (as the number of layers increases, from 32 to 128) for the input image, which has a kernel size of 3x3. The last layer is a convolutional layer connected to a regression layer, which reduces the number of output channels to 3, the same as the number of color channels in the input image.
In addition, intermediate convolutional and batch normalization layers aim to process and denoise the input images effectively by learning complex features and patterns.
Residual Network is used

Training Parameters
The network is trained using the Adam optimizer.
Key parameters include a maximum of 50 epochs, a mini-batch size of 4, and an initial learning rate of 2e-4, with the learning rate being halved every 3 epochs.
Regularization and shuffle options are set to avoid overfitting and ensure robust learning.
Progress plots are enabled for real-time monitoring of the training process.
Early-stop and L2 regularization method are used to avoid overfitting and save the time and computing resources of training.

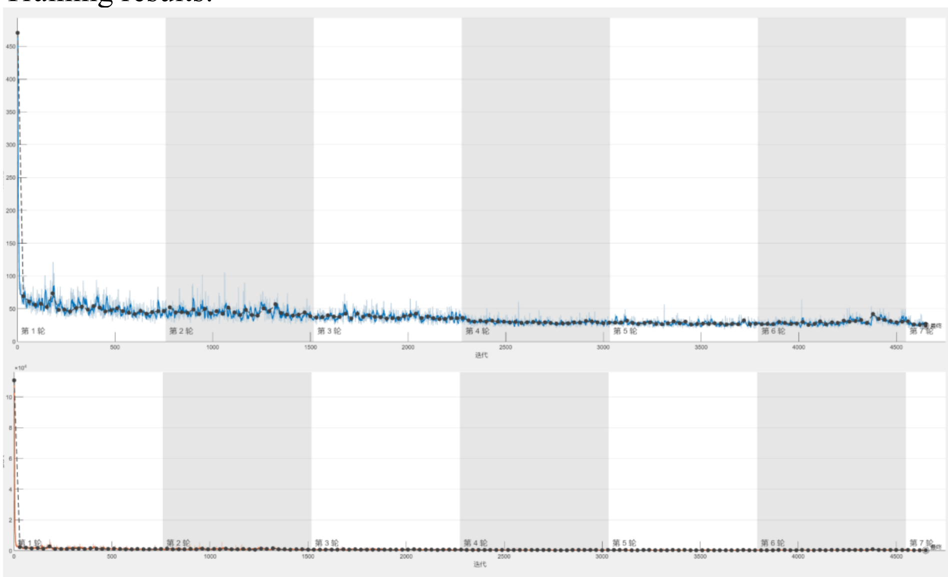
Model Training
The network is trained using the prepared noisy images as input and clean images as the target.
Validation is performed using the separate validation dataset.
The training process involves adjusting the network weights to minimize the difference between the denoised output and the original clean images.
RMSE is calculated as the “accuracy” of the denoising progress.

Saving the Trained Model
Post-training, the model and the validation information are saved for later use in denoising new images in “Custom Denoising Networks” file.

Testing and Evaluation
The trained network is tested on a new set of images (test set).
Noise addition is the same as the training progress.
Performance is evaluated using metrics like Peak Signal-to-Noise Ratio (PSNR) in dB and Structural Similarity Index (SSIM), comparing the outputs of the CNN and some traditional denoised filters (Gaussian, Median, Bilateral), with comprehensive judgment by the box diagram

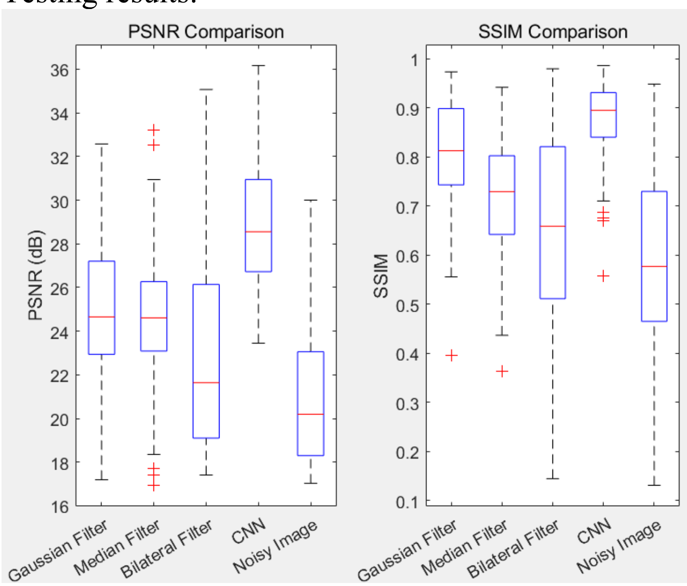
Results & Analysis

Gaussian Noise
Training results:

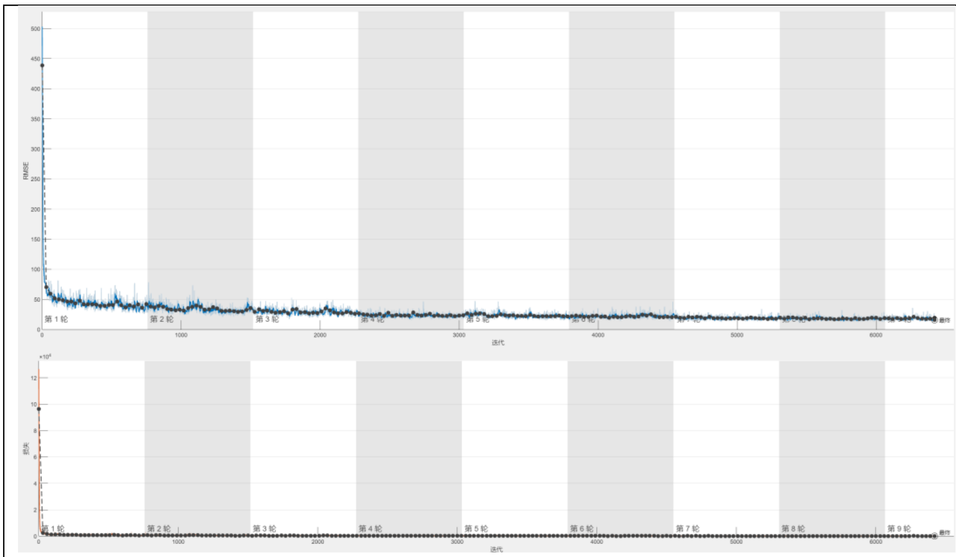


(Figure 1: RMSE Figure 2: Training lost black points: validation results)

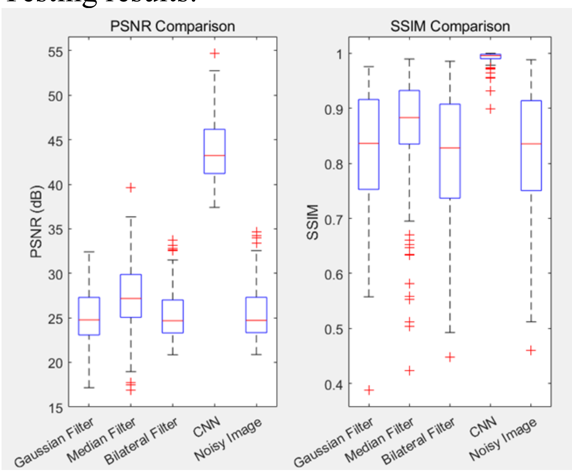
Testing results:



Salt & Pepper Noise
Training results:

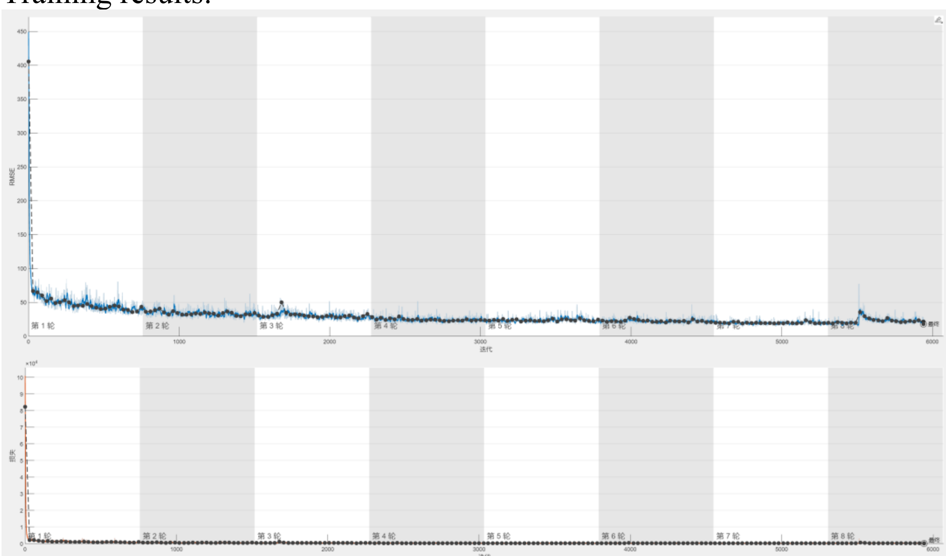


Testing results:

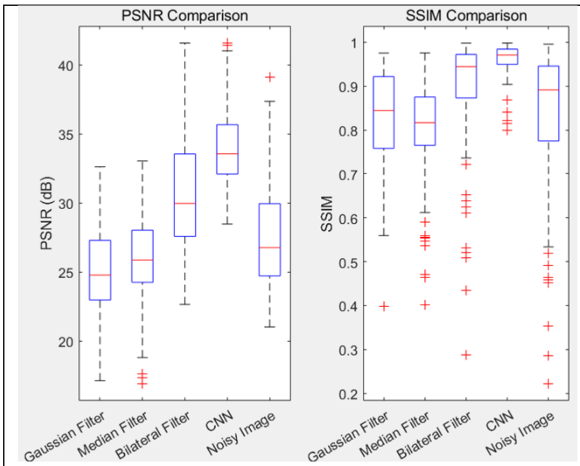


Speckle Noise

Training results:



Testing results:



The CNN consistently outperforms the Gaussian filter in terms of both PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) across all tested noise types with relatively higher and more stable, convergent.
This indicates my CNN's superior ability to reconstruct high-quality images from noisy inputs, preserving both the structural integrity and visual quality, which highlights the advantages of leveraging deep learning for complex image processing tasks.

However, according to my tutor's advice, there are also some parts that I should further improve. The potential method to solve the problem is to further adjust the parameters of the CNN such as the depth of layer, and I will do more training-testing experiments to find better CNN for image denoising.

尚需完成的任务 Work to do:

1. Refine the code to enhance the performance of the Convolutional Neural Networks (CNNs) for a more effective implementation.
2. Train various types of CNN architectures.
3. Compose the final report.

存在问题 Problems:

1. The training results of the CNNs require further improvement.
2. The handling of noise of varying intensities and types needs to be improved.

拟采取的办法 Solutions:

1. Continuously adjust parameters, such as the number of layers, to refine the network structure.
2. Process the noisy image dataset with noise patterns that more closely resemble those found in real-world scenarios, such as K-distribution noise and ultrasound acquisition noise. Moreover, conduct experiments with noise at varying amplitudes.
3. Carry out a broader range of experiments with diverse networks, observe the results, and summarize the findings.

论文结构 Structure of the final report: (Chapter headings and section sub headings)

Supervision log

北京邮电大学 本科毕业设计（论文）教师指导记录表

Project Supervision Log

<table>
<tr><td>学院
School</td><td>International School</td><td>专业
Programme</td><td colspan="3">Telecommunications Engineering with Management</td></tr>
<tr><td>姓
Family name</td><td>Lin</td><td>名
First Name</td><td colspan="3">Guanyi</td></tr>
<tr><td>BUPT 学号
BUPT number</td><td>2020213069</td><td>QM 学号
QM number</td><td>200977962</td><td>班级
Class</td><td>2020215102</td></tr>
<tr><td>论文题目
Project Title</td><td colspan="5">A Study on Image Denoising Using Deep Learning</td></tr>
<tr><td colspan="6">Please record supervision log using the format below:

Date: dd-mm-yyyy
Supervision type: face-to-face meeting/online meeting/email/other (please specify)
Summary:</td></tr>
<tr><td colspan="6">Date: 19-10-2023
Supervision type: email
Summary: discussed the project topic and basic tasks

Date: 23-10-2023
Supervision type: email
Summary: received feedback on my MATLAB code

Date: 30-10-2023
Supervision type: email
Summary: discussed the literature review and how to improve

Date: 08-11-2023
Supervision type: email
Summary: discussed how to train my own CNN on MATLAB

Date: 16-11-2023
Supervision type: email
Summary: received feedback on project specification, and approval to submit

Date: 16-12-2023
Supervision type: email
Summary: received feedback on my literature review,Convolutional Neural Network (CNN) training code on MATLAB and the accompanying lab report

Date: 12-01-2024
Supervision type: email
Summary: received feedback on my Project Early-term Progress Report, and approval to submit

Date: 17-01-2024
Supervision type: face-to-face meeting
Summary: registered in UPV system</td></tr>
</table>

Date: 18-01-2024
Supervision type: face-to-face meeting
Summary: discussed the current status and plans of my work

Date: 24-01-2024
Supervision type: face-to-face meeting
Summary: discussed and introduced the lab

Date: 07-02-2024
Supervision type: face-to-face meeting
Summary: discussed my current version of code, including how to fine-tune the parameters

Date: 13-02-2024
Supervision type: email
Summary: received feedback of my lab report

Date: 23-02-2024
Supervision type: face-to-face meeting
Summary: discussed my Project Mid-term Progress Report and received approval to submit

Date: 26-02-2024
Supervision type: face-to-face meeting
Summary: rehearsed mid-term viva and discussed additional tasks of work in future

Date: 05-03-2024
Supervision type: online meeting
Summary: mid-term viva

Date: 14-03-2024
Supervision type: email
Summary: discussed the improvement of my experiments

Date: 01-04-2024
Supervision type: email
Summary: discussed the results of my experiments

Date: 02-04-2024
Supervision type: face-to-face meeting
Summary: discussed the further work and the final report

Date: 09-04-2024
Supervision type: email
Summary: discussed the draft of final report

Date: 22-04-2024
Supervision type: email
Summary: discussed the improvements of final report

Date: 24-04-2024
Supervision type: email
Summary: discussed the final version of final report and submission

Additional Appendices (as needed)

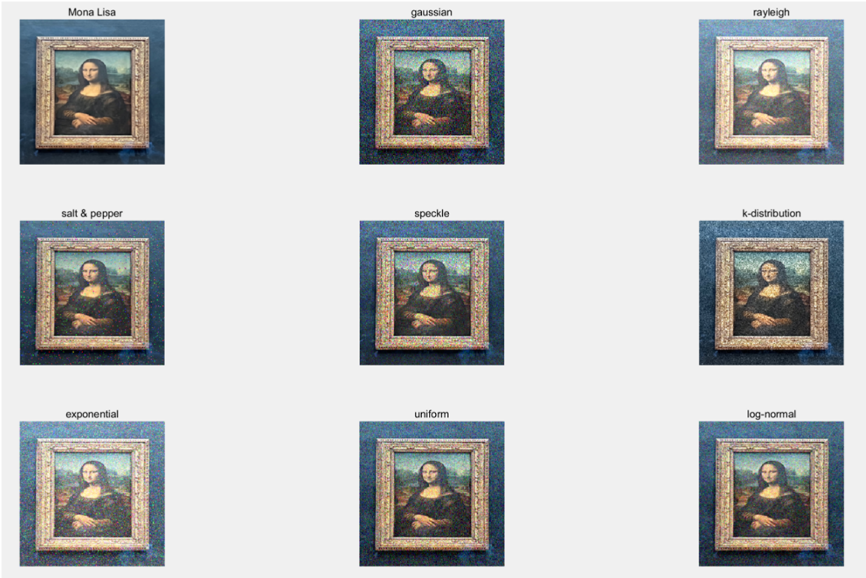


FIGURE 28: An example of different types of image noise

FIGURE 29: An example of image denoising

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.00136	0.01422	0.00321	0.00192	0.00425	**0.00268**	**0.00650**
medium	0.00010	0.00618	0.00526	0.00015	0.00485	**0.00103**	**0.00617**

high	0.00021	0.00848	0.00041	0.00024	0.00440	**0.00022**	**0.00114**

TABLE 22: Comparison of variance of PSNR for Gaussian Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	5.033e-5	3.119e-5	2.082e-5	6.158e-5	1.332e-5	**1.625e-5**	**8.129e-6**
medium	2.939e-5	5.445e-5	6.211e-6	3.029e-5	2.597e-5	**4.234e-5**	**6.418e-5**
high	1.437e-5	6.417e-5	3.313e-5	1.445e-5	4.211e-5	**4.201e-5**	**3.306e-5**

TABLE 23: Comparison of variance of SSIM for Rayleigh Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.00176	0.00700	0.00705	0.00221	0.00229	**0.00340**	**0.01292**
medium	0.00420	0.00468	0.00366	0.00478	0.00416	**0.00169**	**0.01064**
high	0.03149	0.02711	0.02217	0.03175	0.02779	**0.00416**	**0.00561**

TABLE 24: Comparison of variance PSNR for Rayleigh Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	4.164e-5	6.189e-5	3.955e-5	4.051e-5	3.555e-5	**2.369e-5**	**2.846e-6**
medium	7.202e-6	3.44e-5	7.902e-6	1.746e-5	1.892e-5	**1.639e-6**	**2.157e-6**
high	5.639e-5	8.320e-5	8.334e-5	7.995e-5	7.666e-5	**7.774e-6**	**1.595e-6**

TABLE 25: Comparison of variance SSIM for salt & pepper Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.00349	0.02069	0.01703	0.00279	0.00256	**0.00029**	**0.00042**

medium	0.00336	0.00483	0.00263	0.00329	0.00293	**0.00266**	**0.00635**
high	3.544e-5	0.00135	0.00045	3.305e-5	0.00097	**0.00055**	**0.00051**

TABLE 26: Comparison of variance PSNR for salt & pepper Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	1.943e-5	1.708e-5	7.222e-6	1.806e-5	1.188e-5	**3.171e-6**	**8.59e-8**
medium	5.873e-5	1.788e-5	2.153e-6	6.324e-5	4.819e-5	**2.137e-6**	**1.607e-6**
high	9.924e-6	4.199e-5	1.383e-5	1.081e-5	2.698e-5	**9.589e-6**	**1.273e-5**

TABLE : Comparison of variance SSIM for Gaussian Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.00661	0.00268	0.00249	0.00960	0.00670	**0.00456**	**0.00139**
medium	0.02007	0.02173	0.00933	0.02309	0.01311	**0.00363**	**0.00310**
high	0.00092	0.00668	0.00150	0.00122	0.00115	**0.00060**	**0.00065**

TABLE 27: Comparison of variance PSNR for Speckle Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	2.912e-5	3.944e-6	1.180e-5	1.335e-5	2.658e-7	**3.483e-6**	**2.477e-6**
medium	0.00010	4.144e-5	5.660e-5	0.00012	6.671e-6	**5.085e-6**	**7.312e-6**
high	1.181e-7	2.926e-5	2.085e-7	8.422e-7	3.217e-6	**8.026e-8**	**5.616e-7**

TABLE 28: Comparison of variance SSIM for Speckle Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**

low	0.00023	0.00388	0.00645	0.00011	0.00031	**0.00392**	**0.00291**
medium	0.00143	0.00078	0.00120	0.00150	0.00146	**0.00036**	**0.00033**
high	0.00547	0.00542	0.00373	0.00556	0.00619	**0.00096**	**0.00022**

TABLE 29: Comparison of variance PSNR for Exponential Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	8.942e-6	3.276e-5	1.977e-5	1.389e-6	3.719e-6	**2.028e-6**	**1.250e-7**
medium	6.711e-5	6.190e-5	1.116e-5	6.814e-5	5.064e-5	**2.404e-5**	**9.339e-6**
high	9.047e-5	7.495e-5	9.349e-5	0.000110	6.727e-5	**7.718e-5**	**5.275e-5**

TABLE 30: Comparison of variance SSIM for Exponential Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	0.01034	0.00041	0.00784	0.01105	0.00926	**0.00802**	**0.00812**
medium	0.01333	0.00196	0.00440	0.01577	0.00654	**0.00571**	**0.00907**
high	0.00145	0.00189	0.00040	0.00160	0.00342	**0.00020**	**0.00024**

TABLE 31: Comparison of variance PSNR for K-distribution Noise-Affected Images at Different Noise Levels

	Noisy image	Gaussian filter	Median filter	Bilateral filter	DnCNN	**My CNN 5 layers**	**My CNN 10 layers**
low	1.096e-5	4.030e-7	1.676e-5	1.154e-5	1.528e-6	**3.860e-6**	**2.641e-6**
medium	4.753e-5	7.052e-6	2.223e-5	6.296e-5	3.167e-6	**6.066e-6**	**6.350e-6**
high	9.185e-6	3.535e-5	1.965e-6	9.590e-6	3.653e-5	**3.272e-5**	**3.475e-5**

TABLE 32: Comparison of variance SSIM for K-distribution Noise-Affected Images at Different Noise Levels

```
Average PSNR Values: 20.6951      23.4787      24.2766      22.2915      28.1096      27.4521      28.765
Average SSIM Values: 0.57352      0.75509      0.73129      0.6404       0.86903      0.84091      0.88451
图像预处理完成
Average PSNR Values: 16.1647      22.5638      22.2978      16.5158      25.0081      24.2478      24.1676
Average SSIM Values: 0.38225      0.70544      0.61166      0.39631      0.78726      0.72657      0.76014
图像预处理完成
Average PSNR Values: 13.2737      21.2743      20.4781      13.377       22.4907      22.5537      23.403
Average SSIM Values: 0.26355      0.64536      0.50811      0.26694      0.69891      0.65409      0.72084
图像预处理完成
Average PSNR Values: 20.4713      19.2105      20.1359      21.1054      21.0904      29.2903      31.8057
Average SSIM Values: 0.76657      0.73323      0.7695       0.85899      0.8783       0.87243      0.92733
图像预处理完成
Average PSNR Values: 13.6555      13.9741      14.1899      13.8914      14.4156      24.5212      27.4341
Average SSIM Values: 0.52155      0.64073      0.61594      0.57073      0.73847      0.79817      0.85686
图像预处理完成
Average PSNR Values: 9.25353      9.8231       9.59389      9.29837      9.93999      23.0962      23.6487
Average SSIM Values: 0.34119      0.50928      0.43143      0.36014      0.56287      0.71427      0.73749
图像预处理完成
Average PSNR Values: 24.9886      23.8373      26.7955      24.8428      26.3233      38.369       40.9009
Average SSIM Values: 0.82074      0.78578      0.87695      0.80984      0.84089      0.97757      0.99125
图像预处理完成
Average PSNR Values: 19.1864      23.3553      26.6754      19.1852      22.7848      33.4552      36.6716
Average SSIM Values: 0.57478      0.74893      0.87494      0.5663       0.69437      0.95458      0.96948
图像预处理完成
Average PSNR Values: 13.1523      21.1864      25.82        13.1738      20.9747      28.3585      27.9145
Average SSIM Values: 0.28258      0.64123      0.85956      0.2786       0.61172      0.8571       0.87072
图像预处理完成
Average PSNR Values: 25.5156      23.768       25.141       27.9895      30.906       30.2013      31.0852
Average SSIM Values: 0.8003       0.78948      0.79646      0.8613       0.93782      0.91381      0.93502
图像预处理完成
Average PSNR Values: 18.5268      22.9392      22.3655      19.1064      26.1781      26.7163      26.8639
Average SSIM Values: 0.5845       0.75446      0.67457      0.6293       0.86324      0.84994      0.85408
图像预处理完成
Average PSNR Values: 13.0018      20.6353      18.7803      13.1269      21.1697      21.817       23.1546
Average SSIM Values: 0.35419      0.67195      0.51722      0.38078      0.70497      0.70936      0.73759
图像预处理完成
Average PSNR Values: 23.2795      21.5917      23.4445      24.6967      24.9768      30.263       33.3724
Average SSIM Values: 0.76804      0.75692      0.80512      0.85599      0.8984       0.89016      0.9517
图像预处理完成
Average PSNR Values: 17.5468      18.3554      19.7517      18.14        19.6181      27.1163      28.9186
Average SSIM Values: 0.57182      0.70646      0.71293      0.62371      0.80968      0.84805      0.88812
图像预处理完成
Average PSNR Values: 9.8537       11.3128      11.4534      9.91315      11.4661      21.6036      23.3019
Average SSIM Values: 0.29413      0.52327      0.41496      0.31094      0.56387      0.62931      0.70577
图像预处理完成
Average PSNR Values: 23.0726      23.6505      24.4111      24.1987      29.3894      28.3059      29.4812
Average SSIM Values: 0.72656      0.78912      0.76535      0.77418      0.92076      0.88324      0.91061
图像预处理完成
```

FIGURE 30: Examples of testing results

Risk and environmental impact assessment

Please refer to the project handbook section 3.6.12.

There is no risk and environmental impact, because all of the image datasets come from public open-source datasets.