공학박사 학위논문

Prior-adaptive Convolutional Neural Network for Image Denoising

영상 잡음 제거를 위한 프라이어 적응적 컨벌루션 신경망

2017년 8월

서울대학교 대학원

전기 컴퓨터 공학부

안 병 용

ABSTRACT

Digital images suffer from noises that occur throughout the image capturing system. Image denoising is an important process to compensate the noises and plenty of studies have been made in this topic. Due to the ill-posed nature of image denoising, classical studies have developed various image prior models. The denoised image is forced to satisfy some characteristics of nature images that are manually designed. Recently, many researchers have developed discriminative learning based methods. The methods learn a mapping function from a noisy image to the denoised one by a data-driven manner and have shown state-of-the-art performance. In this respect, this dissertation presents convolutional neural network (CNN) based denoising algorithms. While many up-to-date learning based methods minimize only a pixel-based error and miss some important priors as a result, the proposed algorithms are designed to consider priors of natural images and image processing.

First, a self-committee network (SCN) that can find enhanced restoration results from the multiple trial of a trained CNN with different but related inputs is proposed. Specifically, it is noted that the CNN sometimes finds different mapping functions when the input is transformed by a reversible transform and thus produces different but related outputs with the original. Hence averaging the outputs for several different transformed inputs can enhance the results as evidenced by the committee machine methods. Unlike the conventional committee approaches that require several networks, the proposed method needs only a single network. The transformations for constructing a committee are defined from the image processing priors. The SCN method can improve the performance of CNN based denoising methods without additional training or fine-tuning.

A framework to apply an image prior in network training is also presented. Among various prior models, this dissertation employs a non-local self similarity (NSS) prior, which forms a dominant category of image denoising algorithms. The proposed block-matching convolutional neural network (BMCNN) first collects and groups similar patches by block-matching and the denoising network estimates a denoised patch from the corresponding group. Experimental results show that the proposed method is superior to state-of-the-art methods and works well on various types of images. This dissertation also argues that some up-to-date NSS based methods can be interpreted as a kind of the BMCNN structure.

Lastly, the proposed frameworks are applied to non-Gaussian denoising problems. The non-Gaussian noise includes Poisson noise, which is a widely-used artificial noise model, and real-noises from various cameras. Experimental results show that the proposed frameworks work well for various types of noise. Moreover, they show outstanding advantage on real noise, where the noise characteristic is unknown and there exists a misalignment between a noisy image and its reference image. It is expected that the proposed framework enables developments of favorable image enhancement tools for individual real camera.

Key words: Image Denoising, Convolutional Neural Network, Image Prior, Committee, Non-local Self Similarity, Block Matching, Real Noise

Student number: 2011-20871

Contents

Abstract **i**

Contents **v**

List of Figures **ix**

List of Tables **xiii**

1 Introduction **1**

1.1 Self - Committee Network based on Image Processing Prior 4

1.2 Image Denoising by a Block-Matching Convolutional Neural Network 5

1.3 Non-Gaussian Noise 8

1.4 Contribution 9

1.5 Contents 10

2 Related Work **11**

2.1 Prior based Image Denoising 11

2.2 Learning based Image Restoration 13

2.3 Non-Gaussian Image Denoising 15

3 Self-Committee Network based on Image Processing Prior 17

3.1 Committee Machine 17

3.2 Proposed Algorithm 18

3.2.1 Flip and/or Rotation (FR) 19

3.2.2 Linearity 20

3.3 Experimental Results 21

3.3.1 Experiments on image denoising network 22

3.3.2 Experiments on a single image super-resolution network . . . 26

4 Image Denoising by a Block-Matching Convolutional Neural Network 29

4.1 Block Matching Convolutional Neural Network 29

4.1.1 Patch Extraction and Block Matching 30

4.1.2 Network Structure 32

4.1.3 Patch aggregation 35

4.1.4 Connection with Traditional NSS based Denoising 36

4.2 Experiments 39

4.2.1 Training Methodology 39

4.2.2 Training and Test Data 40

4.2.3 Comparision with the State-of-the-Art Methods 41

4.2.4 Effects of Network Formulation 48

5 Applications: Denoising of Non-Gaussian Noise 55

5.1 Denoising of Poisson Noise 55

5.1.1 Poisson Noise 55

5.1.2 Training Criteria 57

5.1.3 Experiments 58

5.2 Denoising of Real Noise 66

5.2.1 Real Noise Database 66

5.2.2 Training Criteria 68

5.2.3 Experiments 69

6 Conclusions **83**

Bibliography **85**

Abstract (Korean) **97**

List of Figures

1.1 A structure of image capturing system with various noise sources. . . 2

1.2 Illustration of (a) noise-free image and (b) corresponding noisy image with $\sigma_n = 25$. 3

1.3 Illustration of (a) neighborhood $S(y_p)$ (in black) of a patch y_p (in red) and (b) similar patches y_{p1}, y_{p2} (in blue). 6

1.4 (a), (b), (c) flowcharts of conventional NSS based system, NN based system and proposed BMCNN respectively and (d) an illustration of block-matching step. 7

3.1 The structure of a general ensemble averaging committee machine. . 18

3.2 The structure of the proposed SCN framework. 18

3.3 The 12 test images used in the experiments 23

3.4 Average IPSNR curves for various SCN structures 23

3.5 The comparison of some features in the original image and inverted image. (a) Input image, (b) features in the first layer, and (c) features in the 13-th layer. 25

3.6 ”Butterfly” image results with their PSNR 28

4.1 The flowchart of proposed BMCNN denoising algorithm. 30

4.2 The architecture of the denoising network 32

4.3 Feature maps from the denoising network. (a) Patches in an input image, (b) the output of the first conv layer, (c) the output of the feature extraction stage (at the same time, the input to the feature processing stage), (d) the output of the feature processing stage (at the same time, the input to the reconstruction stage), (e) the input of the last layer, (f) the output residual patch. 34

4.4 The group denoising schemes of (a) BM3D and (b) WNNM. 37

4.5 Denoising result of the *Fingerprint* image with $\sigma = 25$. 46

4.6 Denoising result of the *Peppers* image with $\sigma = 50$. 46

4.7 The illustration of block matching results for various patch sizes. The first row shows reference patches, the second row shows the 3rd-similar patches to the references and the third row shows the difference of the first and the second row. The error is defined as the average value of the difference. 51

5.1 An example of the Poisson noise 56

5.2 Denoising result of the *Jetplane* with peak value = 1. 59

5.3 Denoising result of the *Fingerprint* with peak value = 5. 60

5.4 Denoising result of the *Barbara* with peak value = 50. 60

5.5 An example of the RENOIR dataset. First row : full images, second row : cropped region of the original images, third row : cropped region of the resized images. The sizes of regions in second row and third row are same. 67

5.6 Denoising results of *scene 36* of T3i dataset. 71

5.7 Denoising results of *scene 40* of T3i dataset. 72

5.8 Denoising results of *scene 39* of S90 dataset. 73

5.9 Denoising results of *scene 40* of S90 dataset. 74

5.10 Denoising results of *scene 36* of Mi3 dataset. 75

5.11 Denoising results of *scene 38* of Mi3 dataset. 76

5.12 Examples of background surface in $S90$ and $T3i$ dataset. 78

5.13 An illustration of the effect of the SCN on T3i-38 scene 81

5.14 An illustration of the effect of the SCN on S90-39 scene 82

List of Tables

3.1 8 FR operations employed to constitute the committee 20

3.2 6 types of committee that are evaluated 22

3.3 Individual PSNR results for gaussian denoising. 24

3.4 Average PSNR results for super-resolution 27

4.1 PSNR of different denoising methods. The best results are highlighted in **bold**. 43

4.2 The average results of WSNR, SSIM, IFC, VIF of various denoising methods with $\sigma = 25$. 47

4.3 Run time (in seconds) of various denoising methods of size 256×256 with $\sigma = 25$. 47

4.4 PSNR results of BMCNN, woBMCNN and their base preprocessing-DnCNN. 49

4.5 PSNR results of BMCNN with various patch sizes. 50

4.6 PSNR and run time results of BMCNN with various stride. 52

4.7 PSNR of BMCNN results with two different preprocessing. 54

5.1 PSNR of different denoising methods. The best results are highlighted in **bold**. 61

5.2 Average PSNR, WSNR, SSIM, IFC, VIF, and run time of different denoising methods for T3i camera. 77

5.3 Average PSNR, WSNR, SSIM, IFC, VIF, and run time of different denoising methods for S90 camera. 77

5.4 Average PSNR, WSNR, SSIM, IFC, VIF, and run time of different denoising methods for Mi3 camera. 78

5.5 Average PSNR, WSNR, SSIM, IFC, and VIF of different denoising methods for various cameras. 80

Chapter 1

Introduction

As high-performance digital devices such as cameras and smartphones are widely available, a huge number of images are captured everyday. Image capturing system of a digital device consists of many processing stages until a digital image is produced from a natural scene. Unfortunately, some stages are usually disturbed by various kinds of noises and the system produces noisy output. Fig. 1.1 shows the pipeline of general image capturing system. Moreover, captured digital images can be affected by additional noises through the transmission and/or storage procedure. Since these noises degrade the visual quality of the image and disturb image understanding or computer vision tasks, image denoising is regarded as an essential step, and plenty of studies [1–32] have been made to solve the problem for decades.

Generally, image denoising is considered as an estimation of the noise-free pixel value $X(i,j)$ from its noisy observation $Y(i,j)$ and the noise is usually assumed additive white Gasussian noise (AWGN):

$$Y(i,j) = X(i,j) + V(i,j) \tag{1.1}$$

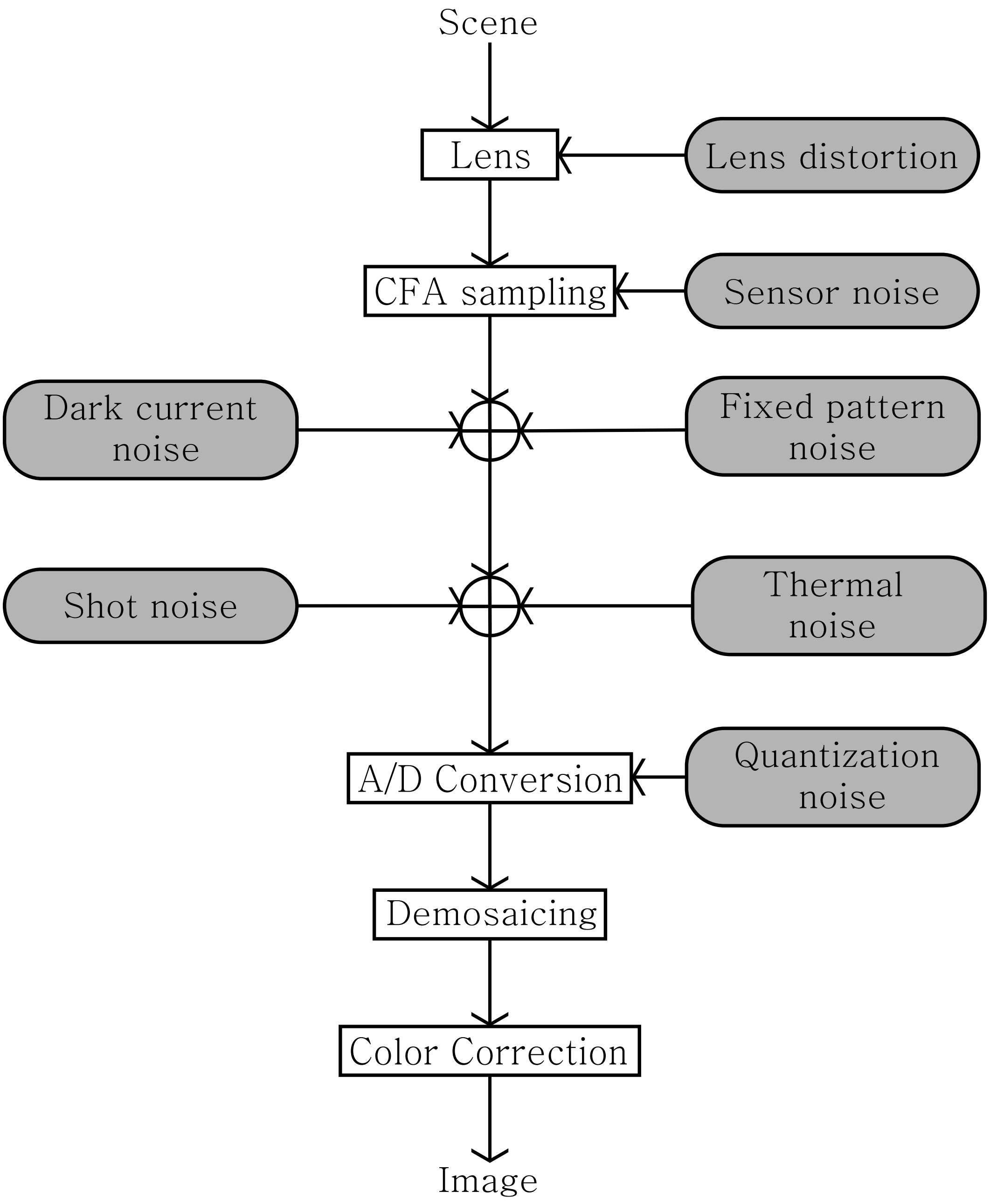
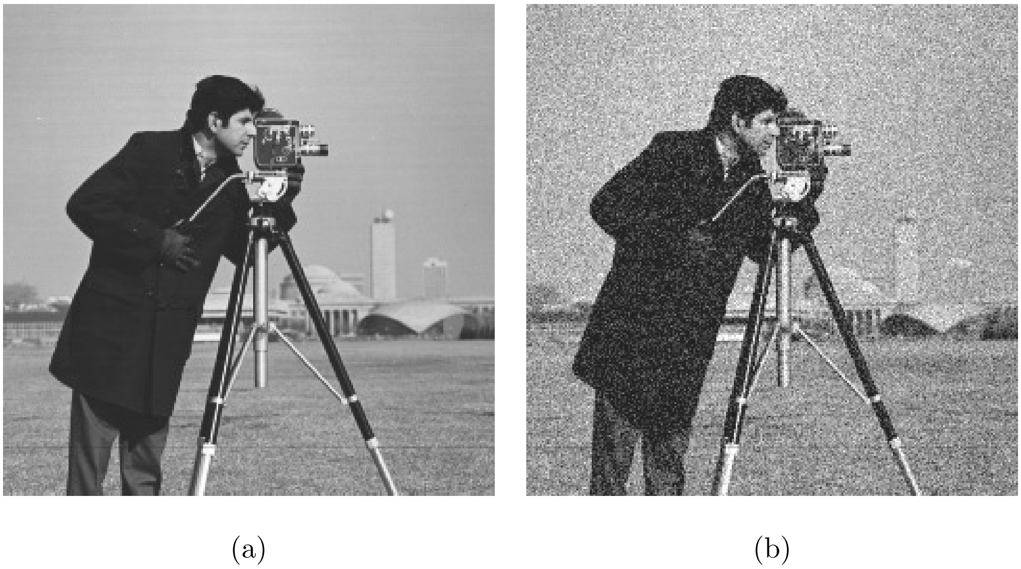


Figure 1.1: A structure of image capturing system with various noise sources.



(a)

(b)

Figure 1.2: Illustration of (a) noise-free image and (b) corresponding noisy image with $\sigma_n = 25$.

$$V(i,j) \sim N(0, \sigma_n^2), i.i.d. \tag{1.2}$$

where σ_n is the standard deviation of noise, named *noise level*. An example of noise-free image and corresponding Gaussian noisy image is shown in Fig. 1.2. The AWGN is regarded as an appropriate model for image noise. Especially, sensor noise caused in low-light environment and Johnson-Nyquist noise in high temperature, which are the dominant parts of real noise, follow Gaussian noise model. Moreover, due to the central limit theorem, it is guaranteed that a sum of many different noises tends to be Gaussian.

In this dissertation, image denoising methods based on convolutional neural network (CNN) structure are proposed. Unlike previous methods that focus on minimizing error function, the proposed approaches develop frameworks to combine prior models with CNN structures. First, a committee machine for a trained network is constructed by transforming the input and output of the network. The transforms are defined from the image processing priors. In addition, a widely-used non-local self similarity prior of a natural image is employed to train a novel denoising network. In order to combine NSS prior with CNN structure, the proposed approach takes a group of similar patches as a input of the network rather than a single patch. This approach is expanded to Poisson noise and real noise from practically captured images.

1.1 Self - Committee Network based on Image Processing Prior

In this dissertation, a committee approach that works at the inference stage to enhance the performance of CNN based image restoration methods is proposed. The idea of "committee machine" for a vision task was introduced in [33, 34], and it was shown to achive the best performance for MNIST digit classification problem [35]. The main idea of this method is to average the outputs of differently trained networks (called member networks) to the same input, which could alleviate the local minima problem and increase the performance. The proposed method differs from the conventional committee approaches in constructing the committee members. Specifically, the proposed method uses only a single network named *base network*, and instead of preparing committees as the different networks, the committees are defined as the outputs of the network with differently transformed inputs. The transforms used for the committee construction are based on the image processing priors that conventional methods satisfy. The trained network sometimes finds different feature map for the transformed input and thus produces different output (when inverse transformed). Thus several transforms are prepared, and the transformed inputs are passed through the network and their outputs are used as committees. The outputs are averaged to be the final output. The proposed method is named self-committee network (SCN) in the sense that only a single network is used. The proposed method can improve the performance of the CNN based image restoration methods without additional training or fine-tuning.

1.2 Image Denoising by a Block-Matching Convolutional Neural Network

In order to solve the ill-posed image denoising problem, many prior models such as gradient model [14, 36, 37], Markov random field (MRF) model [15, 38, 39], and sparse representation model [4, 16, 40] are proposed. Among them, many state-of-the-art methods are based on observation that natural scenes usually contain many repeated patterns. Since Buadas et al. [1] introduced a strategy named nonlocal means filter, plenty of methods [3–5, 7, 16] that estimate ground-truth noise free patch x_p from noisy patch y_p using similar patches in the neighborhood of y_p have been developed. The methods are categorized as nonlocal self-similarity (NSS) based methods and the employment of NSS prior has boosted the performance of image denoising significantly. Fig. 1.3 illustrates the concepts of neighborhood $S(y_p)$ of a patch y_p and similar patches in the neighborhood.

Some researchers, meanwhile, developed neural network based denoising algorithms [2, 9, 30, 31]. They trained networks which take a noisy patch as an input and estimate a noise-free original patch. Since neural network based algorithms are data-driven frameworks, they can learn an optimal processing for local regions pro-

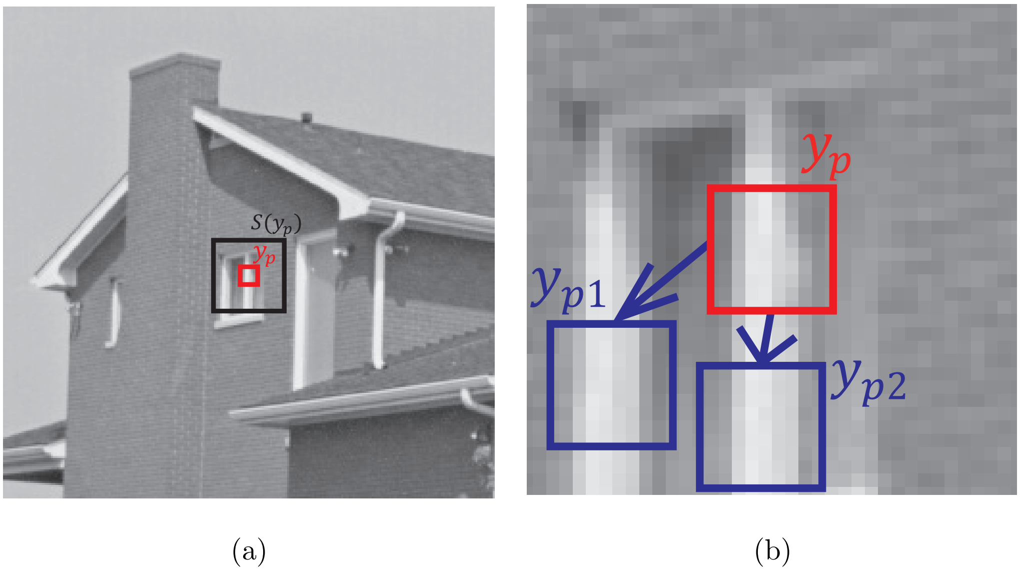
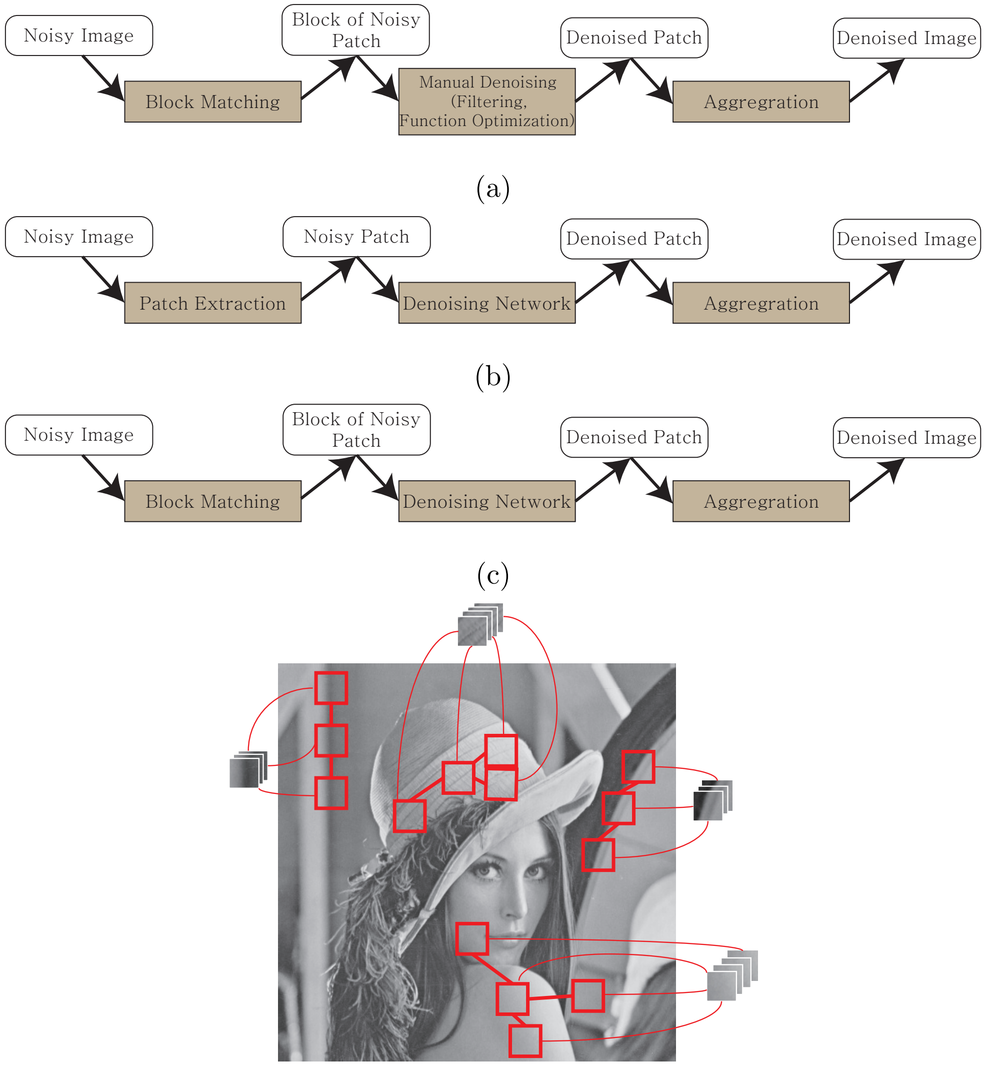


Figure 1.3: Illustration of (a) neighborhood $S(y_p)$ (in black) of a patch y_p (in red) and (b) similar patches y_{p1}, y_{p2} (in blue).

vided that sufficiently large training dataset and deep network are available. The networks can also adopt priors which are not perceptible by the human eye or difficult to implement. Moreover, thanks to the development of graphic processing units (GPU) programming and parallel processing, a neural network (NN) enables much faster denoising than prior model based methods. However, NN based methods tend to show inferior performance to NSS based methods for images with regular and repetitive textures.

In this dissertation, a denoising framework named block-matching convolutional neural network (BMCNN) is presented. The BMCNN is a combined approach of NSS prior and CNN structure. Fig. 1.4 - (a), (b) and (c) show the difference between existing denoising algorithms and the proposed algorithm. As shown in the figures, the proposed method takes a block of similar noisy patches rather than a single patch



(a)

(b)

(c)

(d)

Figure 1.4: (a), (b), (c) flowcharts of conventional NSS based system, NN based system and proposed BMCNN respectively and (d) an illustration of block-matching step.

as an input of denoising network. The block is constructed by block-matching [3] that is illustrated in Fig. 1.4 - (d). Uniting the similar patches enables the network reflecting NSS prior in addition to the local prior that conventional neural networks can train. Compared to the conventional NSS based algorithms, the BMCNN adopts a data-driven framework and consequently achieve the better performance. Moreover, it is shown that some conventional methods can be interpreted as a kind of BMCNN framework.

1.3 Non-Gaussian Noise

Gaussian noise is considered as an appropriate model for image noise. However, in real cases, an image is corrupted by various noise sources and the noise does not necessarily follows the Gaussian distribution. Therefore, denoising the non-Gaussian noise is also an important topic. Especially with high exposures, quantum fluctuations cause noise in the darker part and it is the dominant part of the image noise. This kind of noise is named *shot noise* or *Poisson noise* since it follows the Poisson distribution. In this respect, many up-to-date studies presented denoising algorithms for the Poisson denoising [17–22]. Moreover, a number of mixed noise denoising algorithms [23–27] are also proposed in order to consider both Gaussian noise and Poisson noise consequently.

In this dissertation, it is shown the BMCNN framework achieves favorable performance on non-Gaussian noise denoising. The BMCNN is adopted to two types of noise: Poisson noise and real noise. In order to train and test the network for real noise, this dissertation employs the RENOIR dataset [41] which presents a natural noise dataset from three types of cameras including a mobile phone camera.

Three denoising networks are trained and tested for each camera separately in this dissertation. The training and testing of the denoising network follows the similar procedure to the Gaussian denoising. First, a basic denoising network is trained without a block matching. Then the final network is trained using the outputs from the basic network along with noisy images.

1.4 Contribution

Recently, CNN structure have shown favorable performance on image restoration problems including image denoising. However, up-to-date structures do not consider obvious priors of natural image or image processing and it limits the capacity of the network. For this purpose, this dissertation presents new frameworks to adopt prior models on CNN based denoising. The main contributions of this dissertation are summarized as follows :

- SCN, a committee machine based method is presented. A committee can be constructed based on some image processing priors and the performance is improved.

- BMCNN, a method to combine a NSS prior and CNN structure is proposed. The network works well on both regular and irregular textures.

- The proposed frameworks are expanded to non-Gaussian denoising problems. The network shows robust performance on the real images.

- It is proven that adopting priors of natural image and image processing improves the performance of CNN based denoising.

1.5 Contents

The rest of this dissertation is organized as follows. In chapter 2, researches that are related to the proposed framework are reviewed. There are three main topics: image denoising based on image prior models, learning based image restoration, and non-Gaussian image denoising. Chapter 3 introduces the proposed SCN framework. Its applications with several image restoration algorithms with experimental results and discussions are also presented. In chapter 4, the proposed BMCNN is explained in detail, followed by experimental results and discussion. The proposed frameworks are expanded to the denoising of Poission noise and real noise in chapter 5. Finally, this dissertation is concluded in chapter 6.

Chapter 2

Related Work

2.1 Prior based Image Denoising

Due to the ill-posed nature of image restoration, classical studies have developed various image prior models. Rudin et al. [14] proposed a gradient-projection model that minimizes the total variation of the image. Osher et al. [36] also suggested total-variation problem for image restoration. Their method solved the problem by an iterative regularization method. Weiss et al. [37] derived that derivates of natural images show heavy-tailed non-gaussian distributions. Based on the observation, they designed a maximum-likelihood filter. Chang et al. [28] presented a method to denoise an image in wavelet domain. Remenyi [42] derived the covariance structure of white noise in the wavelet domain. Based on the structure, they extended the wavelet denoising method to the 2D scale-mixing complex wavelet transform domain. Roth et al. [15] proposed field of experts (FoE) method, which models the prior probability of an image in terms of a random field with overlapping cliques. The model is employed as a MRF image priors. Mairai et al. [40] assumed that each image patches can be decomposed into a linear combination of a few elements from a basis set. Based on the assumption, they proposed a sparse coding based image restoration method.

Among the various types of the image prior models, most state-of-the-art denoising methods [1,3,4,7,16] employ nonlocal self-similarity prior of natural images. The nonlocal means filter [1] uses the patch similarity to design a filter for the center pixel of patch. Some other algorithms estimate the denoised patch directly rather than estimating each pixel separately. For the patch denoising, the algorithms adopt another priors of images. In BM3D algorithm [3], some neighborhood patches with high similarity are grouped with the reference patch to form a “group”. The group is transformed to the DCT domain, and the denoising is performed in the transformed domain. Sujeong et al. [12] adopted wavelet prior model and proposed wavelet nonlocal means filter algorithm. Dong et al. [4,5] solved the denoising problem by using the sparsity prior of natural image. Since the matrix formed by similar patches in a nature image is of low rank, the sparse representation of noisy group can be a good solution. Gu et al. [7] also considered denoising as a kind of low-rank matrix approximation problem and introduced weighted nuclear norm minimization (WNNM) algorithm. In addition to the low-rank nature, they took advantage of the prior knowledge that large singular value of the low-rank approximation represents the major components of the image. Therefore the WNNM algorithm adopt the term that prevents large singular values from shrinking in addition to the conventional nuclear norm minimization (NNM) [43] method. Although adopting image prior model have shown promising quality, they suffer from some drawbacks. First, the model is manually designed and they involve parameters that a user needs to fix. Therefore the performance can be limited by human perception. Moreover, the methods find the optimal solution that satisfies the prior by solving a complex optimization problem that yields high computational cost.

2.2 Learning based Image Restoration

Since Lecun et al. [44] showed that their artificial neural network performs very well in digit classification problem, various learning based algorithms have been developed for diverse computer vision problems ranging from low to high-level tasks. Schmidt et al. [45] proposed a cascade of shrinkage fields (CSF). The algorithm unifies the random field-based model and quadratic optimization into a learning frameworks. Chen et al. [29] proposed a trainable nonlinear reaction diffusion (TNRD) model. The model learns parameters for a diffusion model by a gradient descent procedure.

With a dramatic progress of GPU programming and parallel processing, deep learning based methods have also attracted great attention. In the early stage of this work, some multilayer perceptrons (MLP) were adopted for image processing. Burger et al. [2] showed that a plain MLP can compete the state-of-the-art image denoising methods such as BM3D provided that huge training set, deep network and numerous neuron are available. Their method was tested on several type of noise: Gaussian noise, salt-and-pepper noise, compression artifact, etc. Schuler et al. [46] trained the same structure to remove the artifacts that occur from non-blind image deconvolution. Meanwhile, many researchers have developed CNN based algorithms. Jain et al. [9] proposed a CNN for denoising, and discussed its relationship with the MRF model [47]. Dong et al. [48] proposed a SRCNN, which is a convolutional network for image super-resolution. Although their network was lightweight, it achieved superior performance to the conventional non-CNN approaches. They also showed that some conventional super-resolution methods such as sparse coding [49] can be regarded as a special case of deep neural network. Their work was continued to the compression artifact reduction [50]. Kim et al. [51, 52] proposed two algorithms for image super-resolution. In [51], they presented skip-connection from input to output layer. Since the input and output are highly correlated in super-resolution problem, learning their residual results more effective training. In [52], they introduced a network with repeated convolution layers. Their recursive structure enabled a very deep network without huge model and prevented exploding/vanishing gradients [53]. Recently, some techniques such as residual learning [54] and batch normalization [55] have made considerable contributions in developing CNN based image processing algorithms. The techniques contribute to stabilizing the convergence of the network and improving the performance. For some examples, Timofte et al. [56] adopted residual learning for image super-resolution, and Zhang et al. [30] proposed a deep CNN using both batch normalization and residual learning. The network shows state-of-the art performance for many restoration problems including gaussian image denoising, SISR and JPEG image deblocking.

Although deep learning based methods are proven to be effective in many tasks, they are embedding some limitations. First, the training can be struck in a local minima and therefore the initial condition of the training affects the performance. Zhao et al. [57] showed that local minima limits the network performance. Second, since the training aims to minimize only pixel-based error, the network can miss some priors. Burger et al. [31] showed that although data-driven training itself provides promising results, combining some image priors can raise the performance.

2.3 Non-Gaussian Image Denoising

Poisson noise is the most common assumption for non-Gaussian noise. There have been many researches to restore images damaged from Poisson noise. Previous studies can be categorized into 2 categories: Poisson unbiased risk estimate (PURE) based methods and NSS based methods.

Luisier et al. [17] first introduced PURE, which is statistically estimated the mean square error between nature image and the processed noisy image. By minimizing the error with a linear expansion of thresholds (LET) method, the algorithm estimates a favorable denoised image. They also adopted haar thresholding for exact estimation in [18]. Li et al. [58] proposed an image deconvolution algorithm. They parameterized the deconvolution process as a linear combination of LET functions. The parameter is optimized by minimizing PURE.

NSS prior based methods showed promising results in Poisson noise as well as in Gaussian noise. Deledalle et al. [20] extended the NLM algorithm to Poisson denoising problem. Bindilatti et al. [32] also proposed a NLM algorithm. Their method measures patch similarity by stochastic distances instead of the Euclidean distance. Jin et al. [21] designed a nonlocal filter based on weights optimization. Salmon et al. [22] constructed a patch cluster based on NSS prior. They adopted a generalization of principle component analysis(PCA) for the cluster denoising.

Since a single model cannot cover various noise sources in real cases, many studies assumed mixed Poisson-Gaussian noise model. Zhang et al. [23] proposed a multi-scale variance stabilizing transform for mixed process. Foi et al. [27] presented a method to fit the noise model to the RAW data. Luisier et al. [19] extended the PURE-LET method to the denoising of mixed noise. Jezierska et al. proposed some approaches for the blind cases, in which the parameters of noise such as variance is unknown. In [24], they proposed a primal-dual proximal splitting approach based on the data statistics. In [26], they estimated the parameters such as noise variance by an iterative algorithm and their study is developed to the EM approach for noise modeling [25]. Although a mixed noise approximates the major parts of noise well, it is an artificial noise and, it has different characteristics from a real noise. Therefore, Anaya et al. [41] presented a natural noise dataset from 3 types of cameras including a mobile phone camera.

Chapter 3

Self-Committee Network based on Image Processing Prior

3.1 Committee Machine

Committee machine [59] is a system that is composed of multiple neural networks, named *member networks* or *experts*. It uses a divide and conquer strategy in which the outputs of member networks are combined into a single output. Among the existing cases, ensemble averaging which averages the member outputs is the most simple and common approach. The structure of an ensemble averaging machine is illustrated in Fig. 3.1. The resulting committee almost always has improved performance over any single network since grouping can reduce the variance of the network outputs. In some existing methods member networks has different structure each other and in other methods, every member networks share the same structure but they are trained with different initial conditions. In any cases, up-to-date network committee based methods need a number of network models.

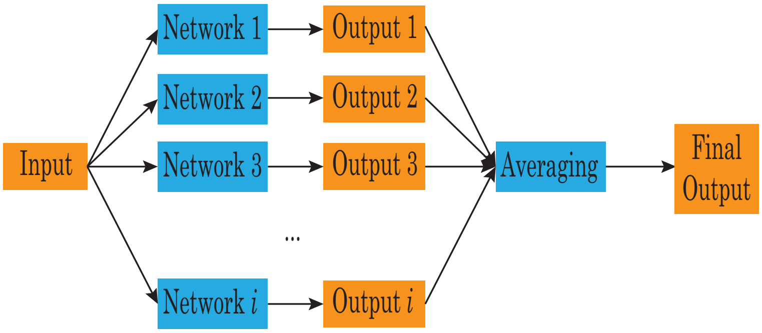


Figure 3.1: The structure of a general ensemble averaging committee machine.

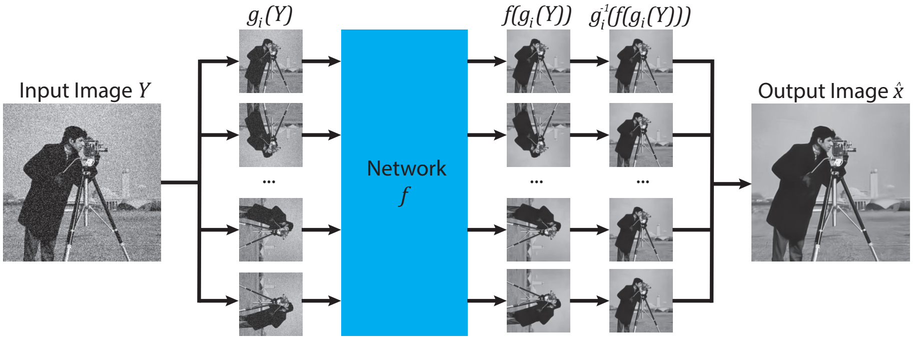


Figure 3.2: The structure of the proposed SCN framework.

3.2 Proposed Algorithm

The key ideas of the proposed method are summarized as follows. Fist, some transformations are applied to an input image, which constructs a group of images for the given input. The group members are individually passed through the network and the outputs are inverse transformed to the original image space. Then the final output is estimated from the group of output images. An example of the proposed SCN framework is illustrated in Fig. 3.2. In this dissertation, two kinds of image transformations are considered, which would bring the output with almost the same performance but different characteristics. Both types of transformations are drawn from the priors of image processing.

3.2.1 Flip and/or Rotation (FR)

Training based image restoration algorithms [2,29,30,45,60] aim to learn a mapping function $f(Y) \sim X$ for a degraded image Y and its ground-truth X. In the view of human visual system (HVS), it is natural that the mapping function should also work the same for the flipped and/or rotated image, i.e., it is desired that the FR image of the restored output must be the same as the restored output of the FR input :

$$f(g(Y)) = g(f(Y)) \tag{3.1}$$

where g is the FR operation. Most prior based image restoration methods satisfy this condition, because the FR operations do not affect the image prior such as gradient distribution or sparsity.

However, it does not hold for the CNN based image restoration methods. Although they augment training data by FR operations [29, 30], it does not force the trained convolution filters to be spatially symmetric, which is needed for FR invariance. Therefore, they produce different results for the FR images and thus it is worth to construct FR committees, where specific operations are summarized in Table. 3.1. In detail, operations $\{g_k\}$ make *member inputs* $\{g_k(Y)\}$ and their corresponding *member outputs* $\{g_k^{-1}(f(g_k(Y)))\}$. The final output is obtained by averaging the member ouputs

;

$$\hat{X}_{FR,I} = \frac{\sum_{k \in K} g_k^{-1}(f(g_k(Y)))}{|K|} \tag{3.2}$$

where K is a subset of $\{1, 2, ...8\}$ and $|K|$ is the size of K.

Table 3.1: 8 FR operations employed to constitute the committee

k	Discription
1	Original
2	FlipUD
3	Rotation (90°)
4	Rotation (90°)+FlipUD
5	Rotation (180°)
6	Rotation (180°)+FlipUD
7	Rotation (−90°)
8	Rotation (−90°)+FlipUD

3.2.2 Linearity

Some image degradation models such as noise-free blurring or image downsampling are assumed as a linear model, $Y = XHV$ where H is a blur kernel and V is a resizing matrix. Therefore, it is natural that their corresponding restoration problems, i.e. deblurring or SISR, are also linear:

$$f(\alpha Y + \beta) = \alpha f(Y) + \beta \tag{3.3}$$

for any scalar α and β.

However, the neural network assumes that the mapping function is non-linear and the network contains bias term in every neuron and non-linear activation functions such as rectified linear unit (ReLU). As a result, (3.3) does not hold for neural network based algorithms, which will produce different outputs for the scaled and/or biased inputs (even when they are restored by removing the bias and rescaled). Hence a committee can be constructed for the member of inputs with several different α and β, i.e., the output is obtained as

$$\hat{X}_L = \frac{\sum_{\alpha,\beta} \hat{x}_{\alpha,\beta}}{\sum_{\alpha,\beta} 1} \tag{3.4}$$

where

$$\hat{x}_{\alpha,\beta} = \frac{f(\alpha Y + \beta) - \beta}{\alpha}. \tag{3.5}$$

However, it is difficult to freely set the α and β in the noisy environments $Y = XHV + N$ where N is the noise, because the scaling α changes the noise characteristics. Assuming that the noise distribution is zero mean and symmetric, just two member networks such that $\{(\alpha, \beta)\} = \{(1, 0), (-1, 1)\}$ are available for the noisy environment in order not to scale the noise component. Specifically, the output is obtained as

$$\hat{X}_I = \frac{f(Y) + (1 - f(1 - Y))}{2} \tag{3.6}$$

which maintain the range of input pixel values, on which the network is trained and works best.

Since the linearity and FR invariance are independent property, they can also cooperate to make a larger committee as

$$\hat{X}_{Full} = \frac{\sum_{\alpha,\beta} \sum_{k \in K} g_k^{-1}(f(g_k(\alpha Y + \beta))) - \beta}{\sum_{\alpha,\beta} \alpha \, |K|} \tag{3.7}$$

3.3 Experimental Results

In this section, experiments for two types of the image restoration are conducted : conducted experiments for two types of the image restoration : Gaussian image denoising and SISR. The performance is evaluated by the peak signal-to-noise ratio

Table 3.2: 6 types of committee that are evaluated

Committee Name	Discription	♯ of Members
SCN-F	Original+Flip ($K = \{1, 2\}$)	2
SCN-R	Original+Rotation ($K = \{1, 3, 5, 7\}$)	4
SCN-FR	Original+FR ($K = \{1 \sim 8\}$)	8
SCN-I	Original+Inversion	2
SCN-Full	Original+FR+Inversion	16
SCN-L	Original+Linear (for SR only)	3

(PSNR) [61] and improved PSNR (IPSNR) compared to the base network. We test 6 types of committees that are summarized in Table. 3.2.

3.3.1 Experiments on image denoising network

For Gaussian image denoising, DnCNN [30] is employed as a base network because of its promising performance and short run-time on GPU. The test set is shown in Fig. 3.3, which is consisted of 12 images that are widely used for the test of image denoising. Fig. 3.4 summarizes the average IPSNR for various noise levels and Table. 3.3 shows the PSNR results on overall test images with $\sigma = 30$.

The results suggest the followings

- The employment of additional committee always improves the performance.
- The information of an image is severely distorted in a high noise level. Therefore, only a single network is hard to be optimal and adding the committees is more beneficial at higher noise level.

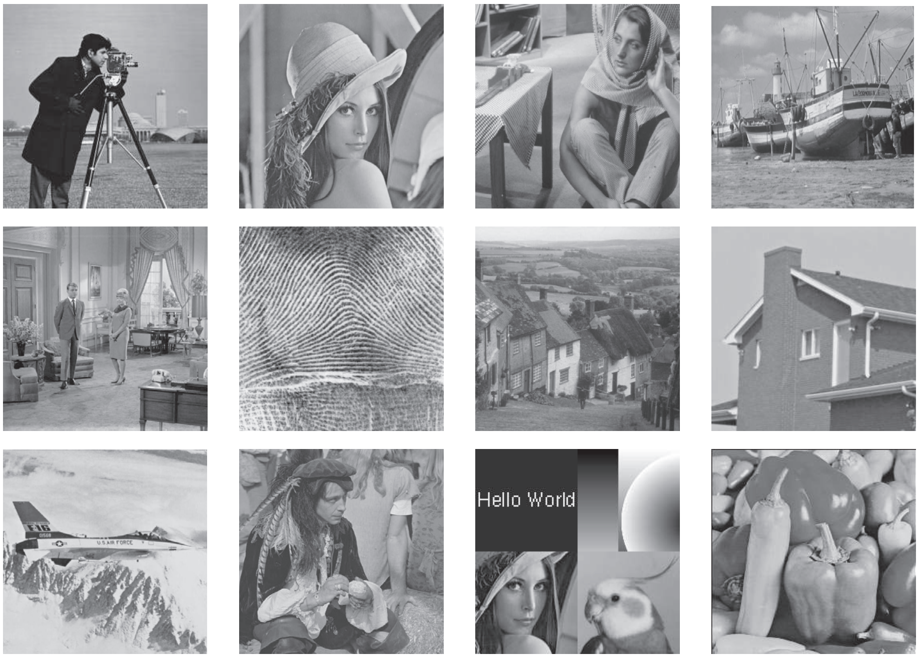


Figure 3.3: The 12 test images used in the experiments

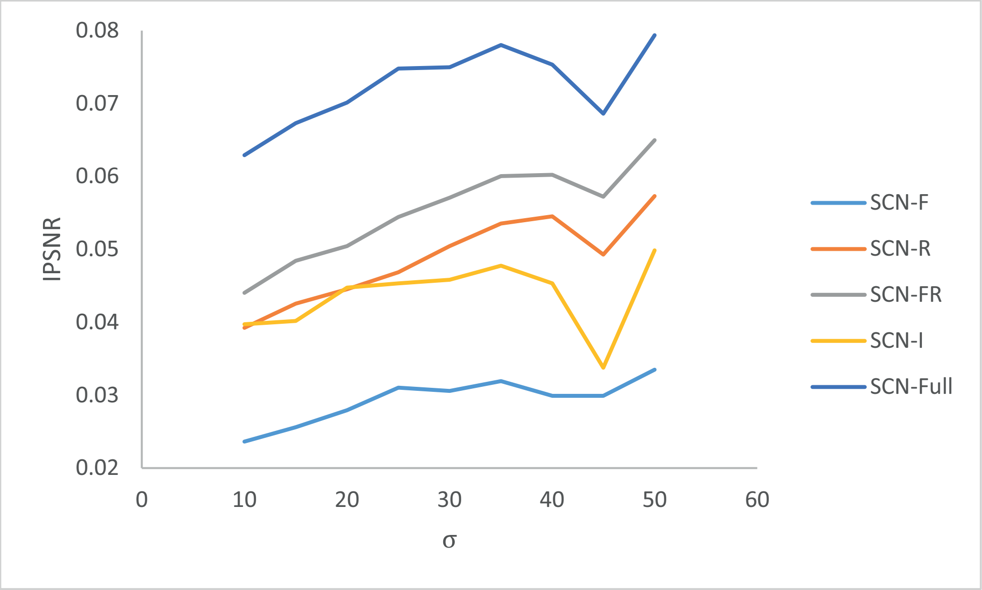


Figure 3.4: Average IPSNR curves for various SCN structures

Table 3.3: Individual PSNR results for gaussian denoising.

Method	DnCNN	SCN-F	SCN-R	SCN-FR	SCN-I	SCN-Full
$\sigma = 30$						
Cameraman	29.24	29.26	29.28	29.28	29.28	29.30
Lena	31.62	31.66	31.67	31.68	31.66	31.69
Barbara	28.84	28.89	28.93	28.94	28.91	28.96
Boat	29.36	29.38	29.40	29.40	29.38	29.41
Couple	29.20	29.22	29.24	29.25	29.23	39.25
Fingerprint	26.61	26.64	26.66	26.67	26.71	26.73
Hill	29.24	29.26	29.26	29.27	29.26	29.27
House	32.38	32.43	32.43	32.44	32.42	33.45
Jetplane	31.12	31.15	31.17	31.17	31.18	31.19
Man	29.23	29.25	29.26	29.27	29.24	29.26
Montage	31.82	31.89	31.93	31.95	31.87	31.94
Peppers	29.86	29.89	29.91	29.91	29.95	29.98
Average	29.87	29.91	29.93	29.94	29.92	29.95

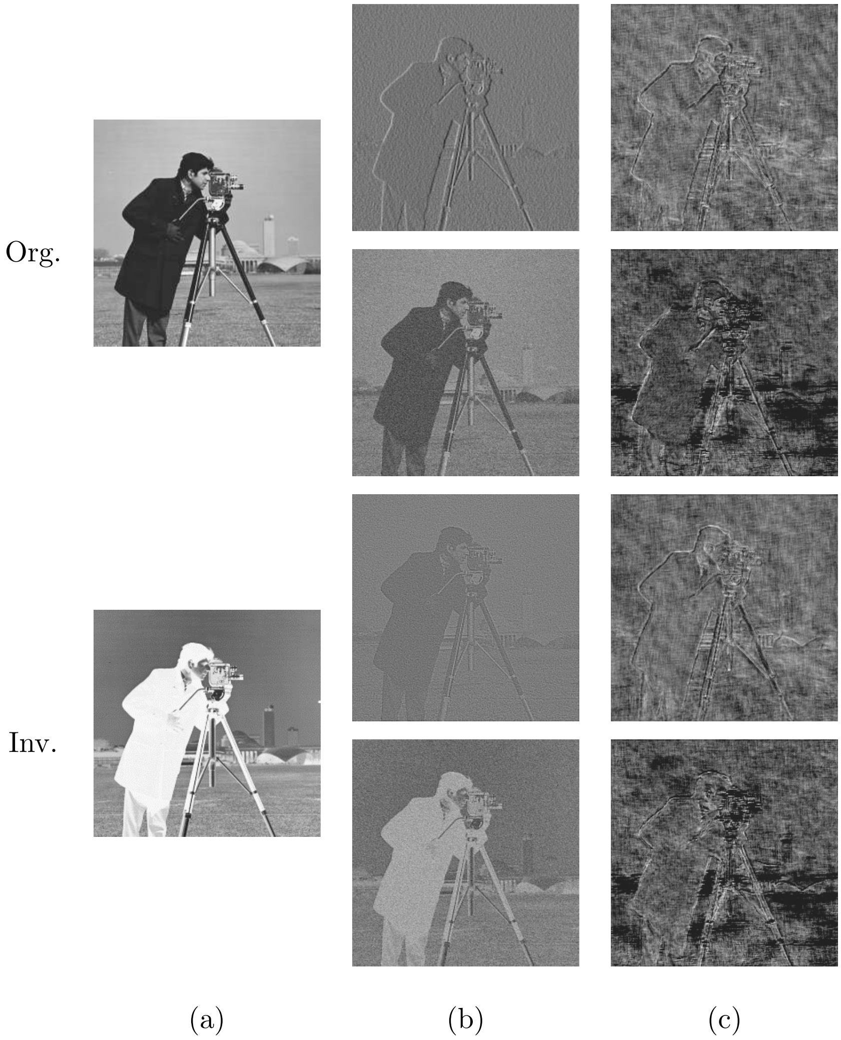


Figure 3.5: The comparison of some features in the original image and inverted image. (a) Input image, (b) features in the first layer, and (c) features in the 13-th layer.

In order to analyze the improvement in view of the feature space, we extracted feature maps from an original image and its inverted one as illustrated in Fig. 3.5. As shown in Fig. 3.5 - (b), low-level feature maps of an inverted image are similar to the inversion of the original feature maps. However, the high-level features show somewhat different characteristics. The original feature map and inverted image feature map are similar in some cases (in the first and third row) but in other cases, they show weak correlation (in the second and fourth row). Moreover, the output of the inverted image would be re-inverted to the original image space and therefore, the two feature maps are distinct in the end. It implies that the function of a committee is expanding the feature maps and enables more accurate process, rather than just augmenting the input.

3.3.2 Experiments on a single image super-resolution network

The proposed SCN framework is also tested for a SISR. In order to show the robustness to the base network, SRCNN [60] is used as a base network. For testing, two test datasets (Set 5 and Set 14) with three scaling factors (2, 3 and 4) are adopted. Four committees as shown in Table. 3.2 are tested: SCN-FR, SCN-I, SCN-L, and SCN-Full. For SCN-L, we set the parameters α and β to

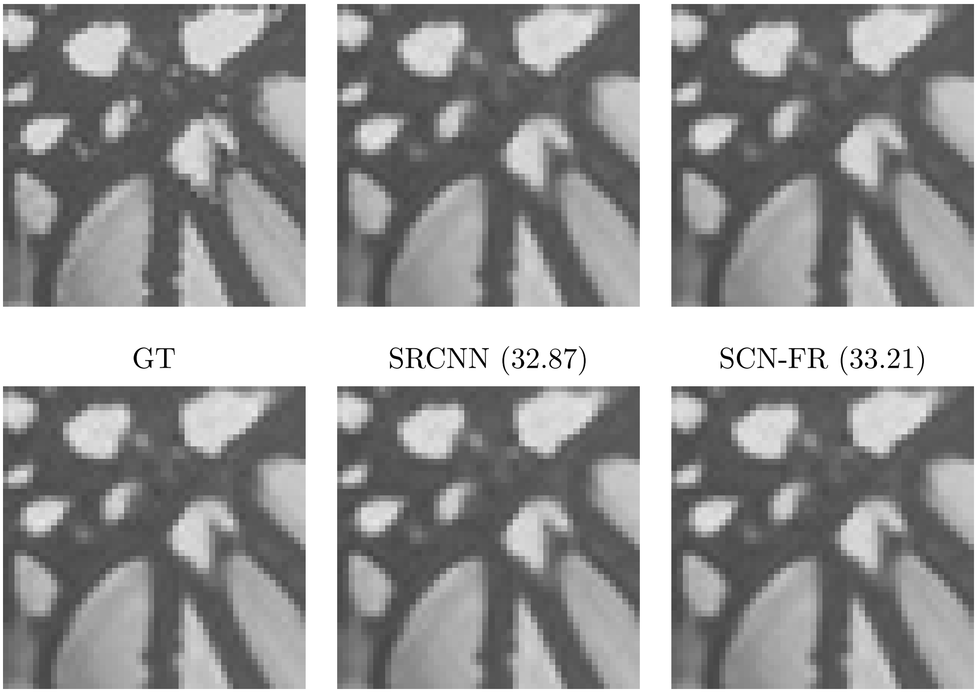
$$\alpha \in \{max(X) - min(X), 1, \tfrac{1}{max(X)-min(X)}\} \tag{3.8}$$

$$\beta = (1-\alpha)mean(X). \tag{3.9}$$

By using these values, the mean pixel value is maintained and the pixel value saturation is prevented. Table 3.4 lists the average PSNRs of different committees and Fig. 3.6 presents an example. As shown in the results, the committee is beneficial for various the image restoration tasks and network formulations. Since the activation function (ReLU) keeps the linearity in a large range, scaling and shifting the input do not show notable difference. On the other hand, the inversion reverses the signs of the feature maps and thus draws out informations that are discarded from the original network. Hence the SCN-I generally yields higher PSNR than the SCN-L, which is just a scaling based committee.

Table 3.4: Average PSNR results for super-resolution

Dataset	Upscaling Factor	SRCNN	SCN-FR	SCN-L	SCN-I	SCN-Full
Set5	2	36.71	36.91	36.72	36.81	36.92
	3	32.83	32.97	32.84	32.89	32.98
	4	30.51	30.63	30.53	30.60	30.64
Set14	2	32.54	32.66	32.55	32.60	32.67
	3	29.34	29.45	29.35	29.39	29.45
	4	27.52	27.58	27.53	27.57	27.59



GT

SRCNN (32.87)

SCN-FR (33.21)

SCN-L (32.97)

SCN-I (33.05)

SCN-Full (33.23)

Figure 3.6: "Butterfly" image results with their PSNR

Chapter 4

Image Denoising by a Block-Matching Convolutional Neural Network

4.1 Block Matching Convolutional Neural Network

In this section, the proposed approach to combine NSS prior and CNN structure for image denoising problem is presented. This section concentrates on additive Gaussian noise model. That is, the proposed BMCNN estimates the original image X from its noisy observation $Y = X + V$, where $V \sim N(0, \sigma^2), i.i.d.$. The overview of BMCNN is illustrated in Fig. 4.1. First, an existing denoising method is applied to the noisy image. The denoised image is regraded as a pilot signal for block matching. That is, the proposed algorithm finds a group of similar patches from the input and pilot images, which is denoised by a CNN. Finally, the denoised patches are aggregated to form the output image.

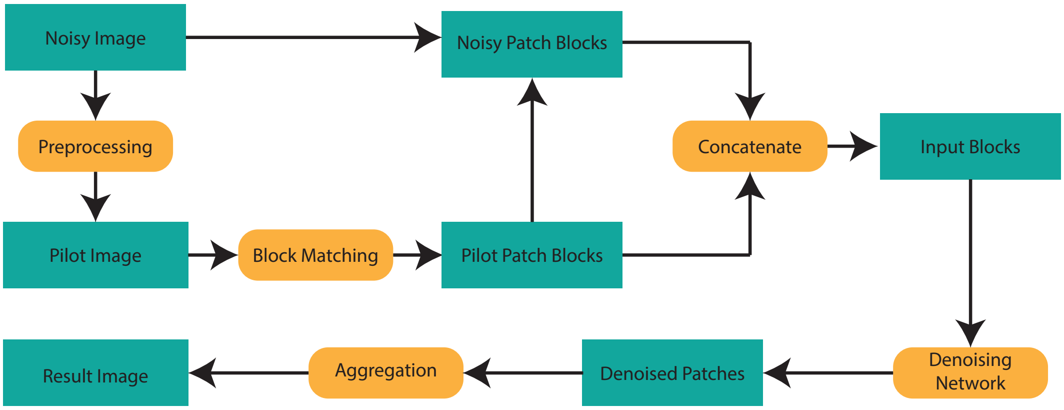


Figure 4.1: The flowchart of proposed BMCNN denoising algorithm.

4.1.1 Patch Extraction and Block Matching

Many up-to-date denoising methods are the patch-based ones, which denoise the image patch by patch. In the patch-based methods, the overlapping patch $\{y_p\}$ of size $N_{patch} \times N_{patch}$ are extracted from Y, centered at the pixel position p. Then, each patch is denoised and merged together to form an output image. In general, this approach yields the best performance when all possible overlapping patches are processed, i.e., when the patches are extracted with the stride 1. However, this is obviously computationally demanding and thus many previous studies [2, 3, 7] suggested to use some larger strides that decrease computations while not much degrading the performance.

In the conventional NSS based algorithm [3], they first find similar patches to y_p based on the dissimilarity measure defined as

$$d(y_p, y_q) = \|y_p - y_q\|^2. \tag{4.1}$$

Specifically, the k patches nearest to y_p including itself are selected and stacked, which forms a 3D block $\{Y_p\}$ of size $N_{patch} \times N_{patch} \times k$. Then the block is denoised in the 3D transform domain. However, it is also shown in [3] that the noise affects the block matching performance too much. Specifically, the distance with the noisy observation is a non-central chi-squared random variable with the mean

$$E(d(y_p, y_q)) = d(x_p, x_q) + 2\sigma^2 N_{patch}^2 \tag{4.2}$$

and variance

$$V(d(y_p, y_q)) = 8\sigma^2 N_{patch}^2 (\sigma^2 + d(x_p, x_q)) \tag{4.3}$$

where x_p and x_q are clean image patches that corresnpond to y_p and y_q respectively. As shown, the variance grows with $O(\sigma^4)$, and thus the block matching results are likely to depend more on the noise distribution as the σ gets larger. This problem is somewhat alleviated by the two-step approach: the first step is to denoise the 3D block as stated above, and the second step is to aggregate the similar patches again by using the denoised patch as a reference. Then the denoised and original patches are stacked together to be denoised again.

In this respect, the proposed algorithm also uses a denoised patch to find its similar patches from the noisy and denoised image. Precisely, an existing algorithm is adopted as a preprocessing step. The preprocessing step finds a denoised image $\hat{X}_{basic}$, which is named *pilot image* and used for the patch aggregation step as follows:

- Block-matching is performed on $\hat{X}_{basic}$. Since the preprocessing attenuates the noise, the block-matching on the pilot image provides more accurate results.

- The group is formed by stacking both the similar patches in the pilot image and the corresponding noisy patches. Since some information can be lost by denoising, noisy input patch can help reconstructing the details of the image.

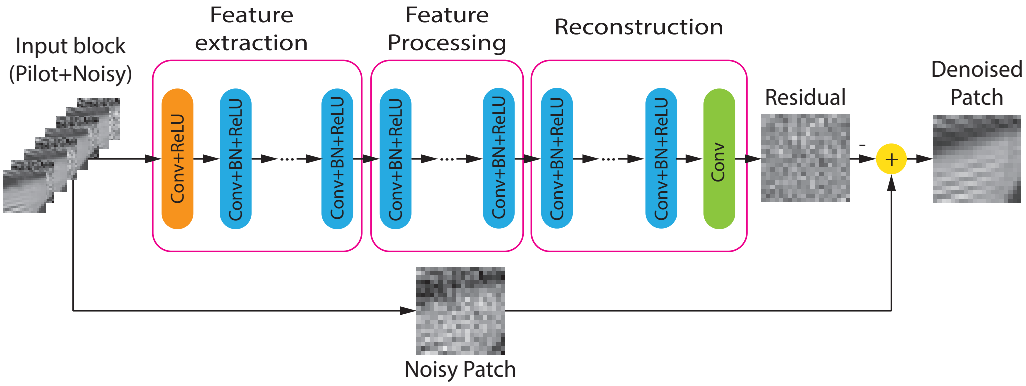


Figure 4.2: The architecture of the denoising network

For most cases, DnCNN [30] is used to find the pilot image because of its promising denoising performance and short run-time on GPU. BM3D algorithm is also used as a preprocessing step, which shows almost the same performance on the average. But the DnCNN and BM3D lead to somewhat different results for the individual image as will be explained in the experiment section.

4.1.2 Network Structure

In CNN based methods, designing a network structure is an essential step that determines the performance. Simonyan et al. [62] pointed out that deep networks consisting of small convolutional filters with the size 3×3 can achieve favorable performance in many computer vision tasks. Based on this principle, the DnCNN [30] employed only 3×3 filters, and the proposed denoising network is also consisted of 3×3 filters, with residual learning and batch normalization. The architecture of the network is illustrated in Fig. 4.2.

In the proposed algorithm, the depth is set to 17, and the network is composed of three types of layers. The first layer generates 64 low-level feature maps using 64 filters of size $3 \times 3 \times 2k$, for the k patches from the input and another k patches from the preprocessed image. Then, the feature maps are processed by a rectified linear unit (ReLU). The layers except for the first and the last layer (layer $2 \sim 16$) contain batch normalization between the convolution filters and ReLU operation. The batch normalization for feature maps is proven to offer some merits in many previous works [55, 63, 64], such as the alleviation of internal covariate shift. All the convolution operations for these layers use 64 filters of size $3 \times 3 \times 64$. The last layer consists of only a convolution layer. The layer uses a single $3 \times 3 \times 64$ filter to construct the output from the processed feature maps. In this dissertation, the network adopts the residual learning, i.e., $f(Y) = V$ [54] . Hence, the output of the last layer is the estimated noise component of the input and the denoised patch is obtained by subtracting the output from the input. These layers can also be categorized into three stages as follows.

Feature Extraction

At the first stage (layer $1 \sim 6$), the features of the patches are extracted. Figs. 4.3(a)$\sim$(c) show the function of the stage. The first layer transforms the input patches into the low-level feature maps including the edges, and then the following layers generate gradually higher-level-features. The output of this stage contains complicated features and some features about the noise components.

Feature Refinement

The second stage (layer $7 \sim 11$) processes the feature maps to construct the target feature maps. In existing networks [48, 50], the refinement stage filters the noise component out because the main objective is to acquire a clean image. On the other

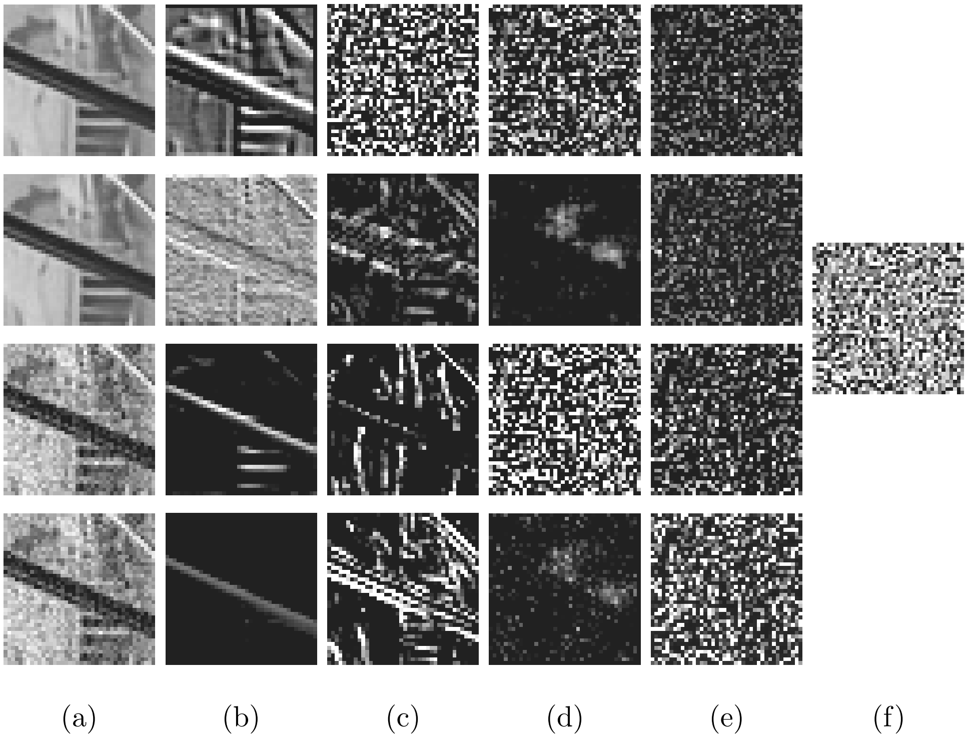


Figure 4.3: Feature maps from the denoising network. (a) Patches in an input image, (b) the output of the first conv layer, (c) the output of the feature extraction stage (at the same time, the input to the feature processing stage), (d) the output of the feature processing stage (at the same time, the input to the reconstruction stage), (e) the input of the last layer, (f) the output residual patch.

hands, the target of the proposed denoising algorithm is a noise patch. Hence, the refined feature maps are comprised of the noise components as shown in Fig. 4.3(d).

Reconstruction

The last stage (layer 12 $\sim$ 17) makes the residual patch from the noise feature maps. The stage can be considered the inverse of the feature extraction stage in that the layers in the reconstruction stage gradually constructs lower-level features from high level feature maps as shown in Figs. 4.3(d)$\sim$(f). Despite all the layers share the similar form, they contribute different operations throughout the network. It gives some intuitions in designing an end-to-end network for image processing.

4.1.3 Patch aggregation

In order to obtain the denoised image, it is straightforward to place the denoised patches $\hat{x}_p$ at the locations of their noisy counterparts y_p. However, as suggested in Sec. 4.1.1, the step size N_{step} is smaller than the patch size N_{patch}, which yields an overcomplete result consequently. In other words, each pixel is estimated in multiple patches. Hence, a patch aggregation step that computes the appropriate value of $\tilde{x}(i,j)$ from a number of estimates $\hat{x}_p(i,j)$ for different p is required. The simplest method for the aggregation is simply taking the mean value of the estimates as

$$\tilde{x}(i,j) = \frac{\sum_{(i,j)\in\hat{x}_p} \hat{x}_p(i,j)}{\sum_{(i,j)\in\hat{x}_p} 1}. \tag{4.4}$$

However, in some studies [2, 46], it is shown that weighting the patches $\hat{x}_p$ with a simple Gaussian window improves the aggregation results. Hence BMCNN also employ the Gaussian weighted aggregation

$$\tilde{x}(i,j) = \frac{\sum_{(i,j)\in\hat{x}_p} w_p(i,j)\hat{x}_p(i,j)}{\sum_{(i,j)\in\hat{x}_p} w_p(i,j)} \tag{4.5}$$

where the weights are determined as

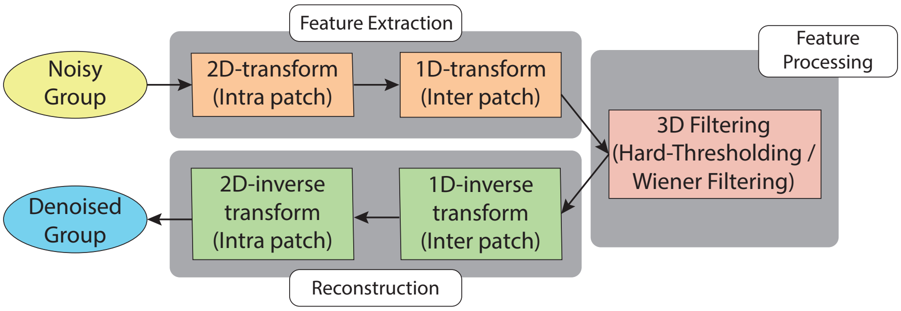
$$w_p(i,j) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp - \frac{|p-(i,j)|^2}{2\sigma_w^2} \tag{4.6}$$

where σ_w is the parameter for weighting.

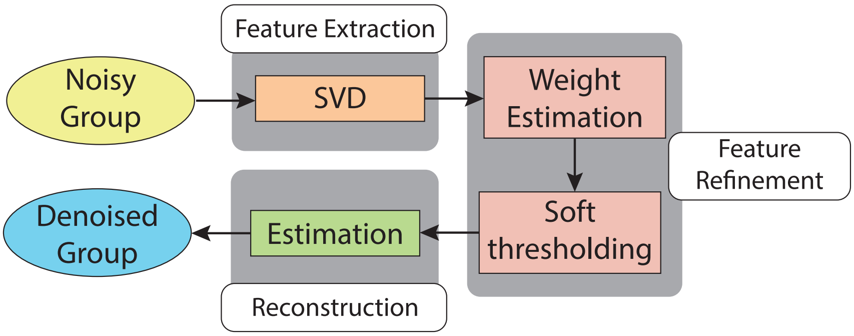
4.1.4 Connection with Traditional NSS based Denoising

In this subsection, it is explained that the proposed BMCNN structure can be considered a generalization of conventional NSS based algorithms. Most existing NSS based denoising algorithms [3–5, 7, 10] share the similar structure: they extract groups of similar patches by block matching, denoise the groups separately, and aggregate the patches to form the output image. Among the procedures, group denoising is the most distinguishing part for individual algorithm. In this sense, the analysis is focused on the group denoising stage. The group denoising stages of state-of-the-art algorithms are illustrated in Fig. 4.4.

In BM3D, the input group is first projected to another domain by a 3D transform. Specifically, they perform a 2D tranform to the patches followed by a 1D transform into the third dimension. Since the transform coefficients are often used as features in many image processing algorithms, the 3D transform corresponds to feature extraction stage. Also, as all the transforms employed in the BM3D (2D discrete cosine transform (DCT), 2D Bior transform and 1D Haar transform) are linear and the parameters are fixed, the entire 3D transform can be considered a convolution network with a single layer of large filter size. In many studies on deep neural network [62, 65], it has been shown that a convolution layer with large receptive field can be replaced by a series of small convolution layers. As a result, the feature extraction operator which involves a series of convolution can be viewed as a


Feature Extraction
Noisy Group
2D-transform (Intra patch)
1D-transform (Inter patch)
Feature Processing
3D Filtering (Hard-Thresholding / Wiener Filtering)
Denoised Group
2D-inverse transform (Intra patch)
1D-inverse transform (Inter patch)
Reconstruction

(a)



(b)

Figure 4.4: The group denoising schemes of (a) BM3D and (b) WNNM.

generalization of the 3D transform. After the transform, a 3D collaborative filtering is applied to the transformed group. Since the filtering attenuates the noise components from each coefficient, it corresponds to the feature refinement operation that maps noisy features to the denoised ones. The BM3D employs hard-thresholding and Wiener filtering for the refinement, where both are one-to-one non-linear mapping. In this sense, the collaborative filtering behaves as a special case of non-linear mapping with 1×1 receptive field that can be implemented using neural network.

WNNM [7] transforms the matrix by singular value decomposition (SVD). The transformation can be viewed as a feature extraction operation that draws some features like basis or singluar value from the matrix. The SVD is relatively a complex operation compared to the multiplication by a constant matrix such as DCT or FFT. In an early work of the neural network [66], however, it has been explained that an optimal solution to an autoencoder is strongly related with the SVD. In other word, the SVD decomposition $[U, \Sigma, V] = SVD(\{Y_p\})$ and the reunion $\{\hat{X_p}\} = U\hat{\Sigma}V^T$ can be successfully replaced by the encoder and decoder of an autoencoder network. Therefore the proposed feature extraction and patch reconstruction operator can be considered a generalization of SVD. The singular values are refined by soft-thresholding whose weights are determined from the singular value itself. Therefore, like the 3D filtering in BM3D, the soft thresholding with the weight estimation behaves as a special case of one-to-one non-linear mapping.

In summary, BMCNN and many NSS based methods follow the same process, i.e., block matching followed by feature extraction and processing in a certain domain by non-linear mappings. However, unlike manually deciding the parameters and non-linear mapping in the existing methods, the proposed BMCNN learns the corresponding procedures and performs denoising in a data-driven manner. It enables finding the optimal or at least suboptimal processing beyond the human design. Furthermore, the BMCNN trains an end-to-end mapping that consists of all operations rather than considering the operations separately.

4.2 Experiments

4.2.1 Training Methodology

The proposed denoising network is implemented using the *Caffe* package [67]. Training a network is to find an optimal mapping function

$$\hat{x}_p = F(\{W_i\}, y_p) \tag{4.7}$$

where W_i is the weight matrix including the bias for the i-th layer. This is achieved by minimizing a cost function

$$L(\{W_i\}) = \frac{1}{N_{sample}} \sum d(\hat{x}_p, x_p) + \lambda r(\{W_i\}) \tag{4.8}$$

where N_{sample} is the total number of the training samples, $d(\hat{x}_p, x_p)$ is the distance between the estimated result $\hat{x}_p$ and its ground truth x_p, $r(\{W_i\})$ is a regularization term designed to enforce the sparseness, and λ is the weight for the regularization term. Zhao et al [57] proposed several loss functions for neural networks, among which we employ the L1 norm for the distance

$$d(\hat{x}_p, x_p) = \sum_k |\hat{x}_p[k] - x_p[k]| \tag{4.9}$$

because of its simplicity for implementation in addition to its promising performance for image restoration. The objective function is minimized using *Adam*, which is known as an efficient stochastic optimization method [68]. In detail, *Adam* solver updates $(W)_i$ by the formula

$$\begin{aligned}(m_t)_i &= \beta_1(m_{t-1})_i + (1-\beta_1)(\bigtriangledown L(W_t))_i, & (4.10)\\ (v_t)_i &= \beta_1(v_{t-1})_i + (1-\beta_1)(\bigtriangledown L(W_t))_i^2, & (4.11)\\ (W_{t+1})_i &= (W_t)_i - \alpha\frac{\sqrt{1-(\beta_2)^t}}{1-(\beta_1)^t}\frac{(m_t)_i}{\sqrt{(v_t)_i}+\epsilon} & (4.12)\end{aligned}$$

where β_1 and β_2 are training parameters, α is the learning rate, and ϵ is a term to avoid zero division. In the proposed algorithm, the parameters are set as: $\lambda = 0.0002$, $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$. The initial values of $(W_0)_i$ are set by Xavier initialization [69]. In *Caffe*, the Xavier initialization draws the values from the distribution

$$(W_0)_i \sim N\left(0, \frac{1}{(N_{in})_i}\right) \tag{4.13}$$

where $(N_{in})_i$ is the number of neurons feeding into the layer. The bias of every convolution layer is initialized to a constant value 0.2. The BMCNN models are trained for three noise levels: $\sigma = 15, 25$ and 50.

4.2.2 Training and Test Data

Recent studies [29, 30] show that less than million training samples are sufficient to learn a favorable network. Following these works, 400 images from the Berkeley Segmentation dataset (BSDS) [70] are used for the training. All the images are cropped to the size of 180 × 180 and data augmentation techniques like flip and rotation are applied. From all the images, training samples are extracted by the procedure in Sec. 4.1.1, i.e., the block size and the stride is set as $20 \times 20 \times 4$ and 20 respectively. The total number of the training samples is 259,200.

The algorithm is tested on standard images that is shown in 3.3. The set contains 4 images of size 256×256 (*Cameraman, House, Peppers* and *Montage*), and 8 images of size 512×512 (*Lena, Barbara, Boat, Fingerprint, Man, Couple, Hill* and *Jetplane*). Note that the test set contains both repetitive patterns and irregularly textured images.

4.2.3 Comparision with the State-of-the-Art Methods

In this section, the performance of the proposed BMCNN is evaluated and compared with the state-of-the-art denoising methods, including NSS based methods (BM3D [3], NCSR [4], and WNNM [7]) and training based methods (MLP [2], TNRD [29], and DnCNN [30]). All the experiments are performed on the same machine - Intel 3.4GHz dual core processor, nVidia GTX 780ti GPU and 16GB memory.

Quantitave and Qualitative Evaluation

The PSNRs of denoised images are listed in Table 4.1. It can be seen that the proposed BMCNN yields the highest average PSNR for every noise level. It shows that the PSNR is improved by 0.1∼0.2dB compared to DnCNN. Especially, there are large performance gains in the case of images with regular and repetitive structure, such as *Barbara* and *Fingerprint*, which are the images that the NSS based methods perform better than the learning based methods. In this sense, it is believed that adopting the patch aggregation brings the advantages of NSS to the learning based method.

In addition to the PSNR, four metrics - structural similarity (SSIM) [71], weighted SNR (WSNR) [72], information fidelity criterion (IFC) [73] and visual information fidelity (VIF) [74] based on the diverse HVS prior are also adopted to analyze the performance from various aspects. Table 4.2 summarizes the average metrics for $\sigma = 25$. Since the training aims to minimize the error, learning based DnCNN shows superior performance to prior based WNNM in WSNR and SSIM, which are based on the difference and correlation respectively. WNNM, on the other hands, shows advantages in information based IFC and VIF. Since prior based methods are inclined to recover general image parts such as edges, it can preserve more informations than learning based methods. The proposed BMCNN combines the prior and training and therefore, shows the best performance in every metrics. Figs. 4.5 and 4.6 illustrate the visual results. The NSS based methods tend to blur the complex parts like the stalk of a fruit and the learning based methods often miss details on the repetitive parts such as the stripes of fingerprint. In contrast, the BMCNN recovers the textures in both types of regions.

Table 4.1: PSNR of different denoising methods. The best results are highlighted in **bold**.

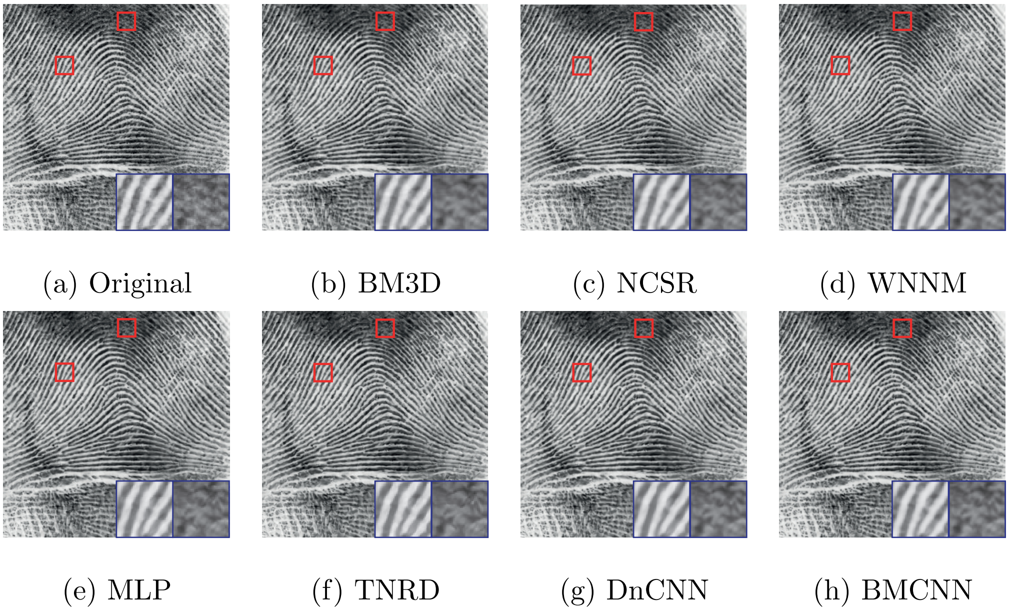
Method	BM3D	NCSR	WNNM	MLP	TNRD	DnCNN	BMCNN
$\sigma = 15$							
Cameraman	31.91	32.01	32.17	-	32.18	32.61	**32.73**
Lena	34.22	34.11	34.35	-	34.23	34.59	**34.61**
Barbara	33.07	33.03	**33.56**	-	32.11	32.60	33.08
Boat	32.12	32.04	32.25	-	32.14	32.41	**32.42**
Couple	32.08	31.94	32.13	-	31.89	32.40	**32.41**
Fingerprint	30.30	30.45	**30.56**	-	30.14	30.39	30.41
Hill	31.87	31.90	32.00	-	31.89	**32.13**	32.08
House	35.01	35.04	**35.19**	-	34.63	35.11	35.16
Jetplane	34.09	34.11	34.38	-	34.28	**34.55**	34.53
Man	31.88	31.92	32.07	-	32.18	**32.42**	32.39
Montage	35.11	34.89	35.65	-	35.02	35.52	**35.97**
Peppers	32.68	32.65	32.93	-	32.96	33.21	**33.32**
Average	32.86	32.84	33.10	-	32.82	33.16	**33.26**

$\sigma = 25$

Cameraman	29.44	29.47	29.64	29.59	29.69	30.11	**30.20**
Lena	32.06	31.95	32.27	32.28	32.05	32.48	**32.53**
Barbara	30.64	30.57	**31.16**	29.51	29.33	29.94	30.58
Boat	29.86	29.68	30.00	29.94	29.89	30.21	**30.25**
Couple	29.69	29.46	29.78	29.72	29.69	30.10	**30.12**
Fingerprint	27.71	27.84	27.96	27.66	27.33	27.64	**28.01**
Hill	29.82	29.68	29.96	29.83	29.77	29.99	**30.00**
House	32.95	32.98	**33.33**	32.66	32.64	33.23	33.32
Jetplane	31.63	31.62	31.89	31.87	31.77	32.06	**32.17**
Man	29.56	29.56	29.73	29.83	29.81	**30.06**	30.06
Montage	32.34	31.84	32.47	32.09	32.27	32.97	**33.47**
Peppers	30.21	29.96	30.45	30.45	30.51	30.80	**30.93**
Average	30.49	30.38	30.72	30.44	30.39	30.80	**30.97**

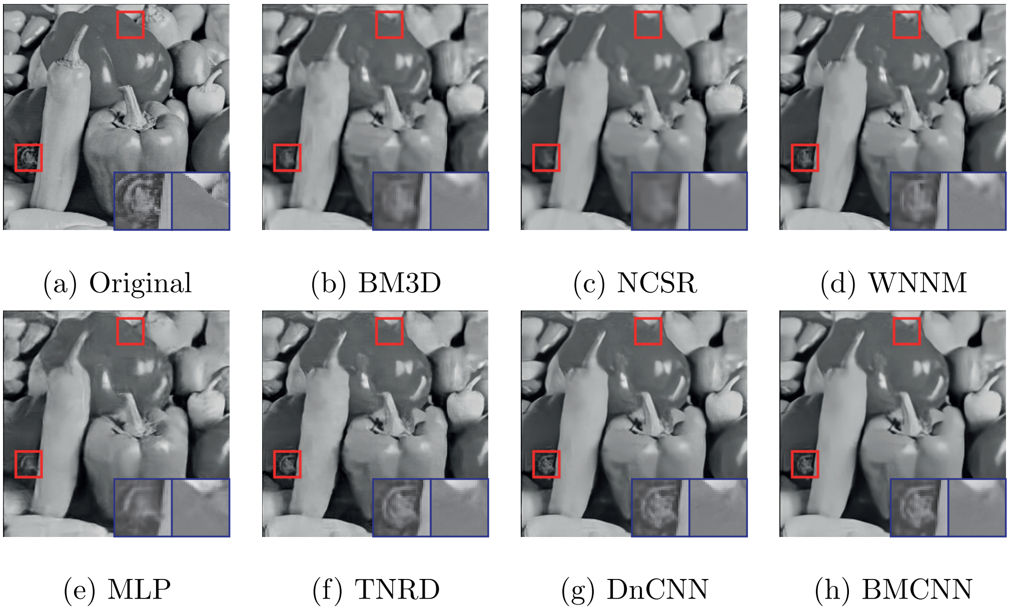
$\sigma = 50$

Cameraman	26.18	26.15	26.47	26.37	26.56	26.99	**27.02**
Lena	29.05	28.97	29.32	29.28	28.94	29.42	**29.56**
Barbara	27.08	26.93	**27.70**	25.26	25.69	26.13	26.84
Boat	26.72	26.50	26.89	27.04	26.85	27.17	**27.19**
Couple	26.42	26.19	26.59	26.68	26.48	26.88	**26.91**
Fingerprint	24.55	24.52	**24.79**	24.21	23.70	24.14	24.65
Hill	27.05	26.87	27.12	**27.37**	27.11	27.31	27.33
House	29.70	29.69	30.25	29.82	29.40	30.08	**30.25**
Jetplane	28.31	28.23	28.61	28.56	28.43	28.74	**28.88**
Man	26.73	26.62	26.91	27.05	26.94	**27.18**	27.17
Montage	27.65	27.62	27.97	28.06	28.12	29.03	**29.50**
Peppers	26.69	26.64	26.97	26.71	27.05	27.30	**27.45**
Average	27.18	27.08	27.47	27.20	27.11	27.53	**27.73**


(a) Original (b) BM3D (c) NCSR (d) WNNM

(e) MLP (f) TNRD (g) DnCNN (h) BMCNN

Figure 4.5: Denoising result of the *Fingerprint* image with $\sigma = 25$.


(a) Original (b) BM3D (c) NCSR (d) WNNM

(e) MLP (f) TNRD (g) DnCNN (h) BMCNN

Figure 4.6: Denoising result of the *Peppers* image with $\sigma = 50$.

Table 4.2: The average results of WSNR, SSIM, IFC, VIF of various denoising methods with $\sigma = 25$.

Methods	BM3D	NCSR	WNNM	MLP	TNRD	DnCNN	BMCNN
WSNR	39.82	39.58	40.02	39.88	39.73	40.15	**40.26**
SSIM	0.8632	0.8610	0.8673	0.8653	0.8618	0.8718	**0.8738**
IFC	2.465	2.427	2.532	2.452	2.397	2.520	**2.597**
VIF	0.4350	0.4486	0.4541	0.4296	0.4174	0.4446	**0.4555**

Table 4.3: Run time (in seconds) of various denoising methods of size 256×256 with $\sigma = 25$.

Methods	BM3D	NCSR	WNNM	MLP	TNRD	DnCNN	BMCNN
256 × 256	0.87	190.3	179.9	2.238	0.038	0.053	2.135
512 × 512	3.77	847.9	778.1	7.797	0.134	0.203	8.030

Run Time

Table 4.3 shows the average run-times of the denoising methods for the images of sizes 256 × 256 and 512 × 512. For TNRD, DnCNN and BMCNN, the times on GPU are computed. As shown, many conventional NSS based methods need very long times, which is mainly due to the complex optimization and/or matrix decomposition. On the other hand, since the BM3D consists of simple linear transform and non-linear filtering, it is much faster than the NCSR and WNNM. Since the BMCNN also consists of convolution and simple ReLU function, its computational cost is also less than the WNNM and NCSR. The BMCNN is, however, slower than other learning based approaches for three main reasons. First, proposed algorithm is a two-step approach that uses another end-to-end denoising algorithm as a preprocessing step. Therefore the computational cost is doubled. Second, the BMCNN contains a block matching step, which is difficult to be implement with GPU. In the proposed algorithm, the block matching step takes almost half of the run-time. Finally, the BMCNN is a patch-based algorithm with the stride less than the patch size. Hence, a pixel is processed multiple times and the overall run-time increases. Although the proposed algorithm is slower for these reasons, it is still competitive considering that it is much faster than the conventional NSS based methods and that it provides higher PSNR than others.

4.2.4 Effects of Network Formulation

In this subsection, some settings of the BMCNN are modified to investigate the relations between the settings and performance. All the additional experiments are made with $\sigma = 25$.

NSS Prior

In this section, the main idea is to take the NSS prior into account by adopting the block matching, i.e., by using the aggregated similar patches as the input to the CNN. In order to show the effect of NSS prior, an additional experiment is made: a network that estimates a denoised patch using only two patches as the input , specifically a noisy patch and the corresponding pilot patch without further aggregation is trained. The network is named as woBMCNN, and its PSNR results are summarized in Table. 4.4. The result validates that the performance gain is owing to the block matching rather than two-step denoising. Interestingly, the woBMCNN does not perform better than DnCNN, which is used as the preprocessing. Actually, the woBMCNN can be interpreted as a deeper network with similar network formulation and a skip connection [75]. However, since the DnCNN is already a favorable network and the performance with the formulation is saturated, the deeper network can hardly perform better. On the other hands, the BMCNN encodes additional information to the network, which is shown to play an important role.

Table 4.4: PSNR results of BMCNN, woBMCNN and their base preprocessing-DnCNN.

	BMCNN	woBMCNN	DnCNN
Cameraman	30.20	30.08	30.11
Lena	32.53	32.45	32.48
Barbara	30.58	29.91	29.94
Boat	30.25	30.17	30.21
Couple	30.12	30.09	30.10
Fingerprint	28.01	27.61	27.64
Hill	30.00	29.96	29.99
House	33.32	33.20	33.23
Jetplane	32.17	32.02	32.06
Man	30.06	30.04	30.06
Montage	33.47	33.02	32.97
Peppers	30.93	30.81	30.80
Average	30.97	30.78	30.80

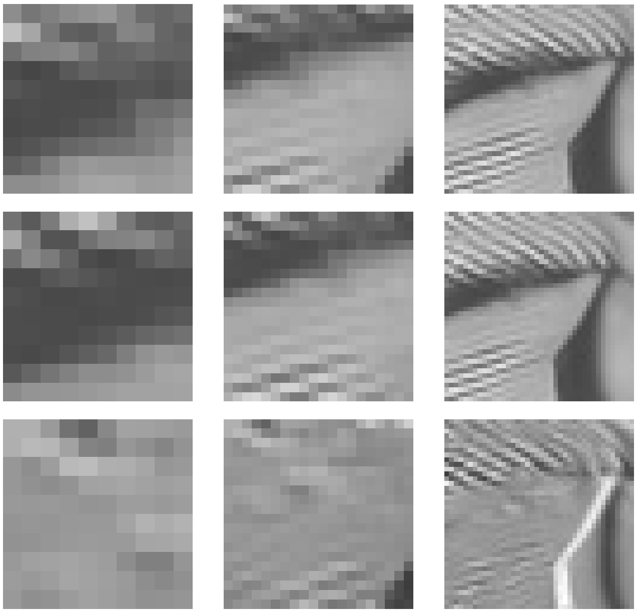
Table 4.5: PSNR results of BMCNN with various patch sizes.

	10 × 10	20 × 20	40 × 40
Cameraman	28.82	30.20	30.00
Lena	30.42	32.53	32.39
Barbara	29.05	30.58	29.72
Boat	29.01	30.25	30.07
Couple	28.89	30.12	29.97
Fingerprint	27.24	28.01	27.51
Hill	28.83	30.00	29.89
House	30.85	33.32	33.11
Jetplane	30.20	32.17	31.96
Man	28.84	30.06	29.96
Montage	30.47	33.47	32.72
Peppers	29.31	30.93	30.67
Average	29.33	30.97	30.66

Patch Size

In many patch-based algorithms, the patch size is an important parameter that affects the performance. In this sense, this dissertation trains three networks with patch size 10 × 10, 20 × 20 (base) and 40 × 40. Table 4.5 shows that 20 × 20 patch works better than other sizes. Moreover, networks of patch size 10 × 10 and 40 × 40 work even worse than its preprocessing.

Burger et al. [2] revealed that a larger patch contains more information, and thus the neural network can learn more accurate objective function with larger training



(a) 10 × 10 (b) 20 × 20 (c) 40 × 40

Error : 0.0515 Error : 0.0672 Error : 0.1813

Figure 4.7: The illustration of block matching results for various patch sizes. The first row shows reference patches, the second row shows the 3rd-similar patches to the references and the third row shows the difference of the first and the second row. The error is defined as the average value of the difference.

patches. On the contrary, it is difficult to train the mapping function reasonably with small patches. But the large patch degrades the block matching performance, because it becomes more difficult to find well matched patches as the patch size increases. Fig. 4.7 shows the block matching result and the error for several patch sizes. For the 10×10 and 20×20 patches, the block-matching finds almost the same patches and the error is very small. However in the case of 40×40 patch, decent portion of the patch does not fit well and the error becomes so big. Conventional

NSS based algorithms including [3] and [5] also prefer small patches whose sizes are around 10 × 10 for these reasons. To conclude, 20 × 20 is considered the proper patch size that satisfies both CNN and NSS prior.

Table 4.6: PSNR and run time results of BMCNN with various stride.

	5	10	15	20
Cameraman	30.21	30.20	30.20	30.18
Lena	32.53	32.53	32.52	32.51
Barbara	30.58	30.58	30.56	30.49
Boat	30.25	30.25	30.25	30.24
Couple	30.12	30.12	30.12	30.11
Fingerprint	28.03	28.01	28.01	27.97
Hill	30.00	30.00	30.00	30.00
House	33.32	33.32	33.32	33.30
Jetplane	32.17	32.17	32.16	32.16
Man	30.06	30.06	30.05	30.05
Montage	33.49	33.47	33.45	33.40
Peppers	30.94	30.93	30.93	30.92
Average PSNR	30.98	30.97	30.96	30.94
Average time(256 × 256)	4.271	2.151	1.765	1.603
Average time(512 × 512)	16.79	8.101	6.492	5.777

Stride

Since the proposed method is patch-based, its performance depends on the stride value to divide the input images into the patches. With a small stride, each pixel appears in many patches, which means that every pixel is processed multiple times. It definitely increases computational costs but has the possiblity of performance improvement. The proposed BMCNN is tested with various stride values and the results are summarized in the Table 4.6. From the result, it is determined that a stride value around the half of the patch size shows reasonable performance for both the run time and the PSNR.

Pilot Signal

In this subsection, an experiment is conducted to show how the different preprocessing methods (other than DnCNN in the previous experiments) affect the the overall performance. For the experiment, BM3D [3] algorithm is employed due to its NSS based nature and reasonable run time. Table 4.7 shows the denoising performance with different preprocessing methods and two interesting characteristics can be found.

- The performance on the individual image depends on the preprocessing method. BMCNN-BM3D shows better performance on *Barbara, Fingerprint* and *House*, where NSS based WNNM performed better than the CNN based DnCNN.
- However, the overall performance shows negligible difference. It implies the overall performance of the denoising network depends on the network formulation, not the preprocessing.

Table 4.7: PSNR of BMCNN results with two different preprocessing.

	BMCNN-DnCNN	BMCNN-BM3D
Cameraman	**30.20**	30.03
Lena	**32.53**	32.49
Barbara	30.58	**31.23**
Boat	**30.25**	30.16
Couple	**30.12**	30.02
Fingerprint	28.02	**28.06**
Hill	30.00	**30.05**
House	33.32	**33.43**
Jetplane	**32.17**	32.14
Man	**30.06**	29.94
Montage	**33.47**	33.31
Peppers	**30.93**	30.78
Average	**30.97**	30.97

Chapter 5

Applications: Denoising of Non-Gaussian Noise

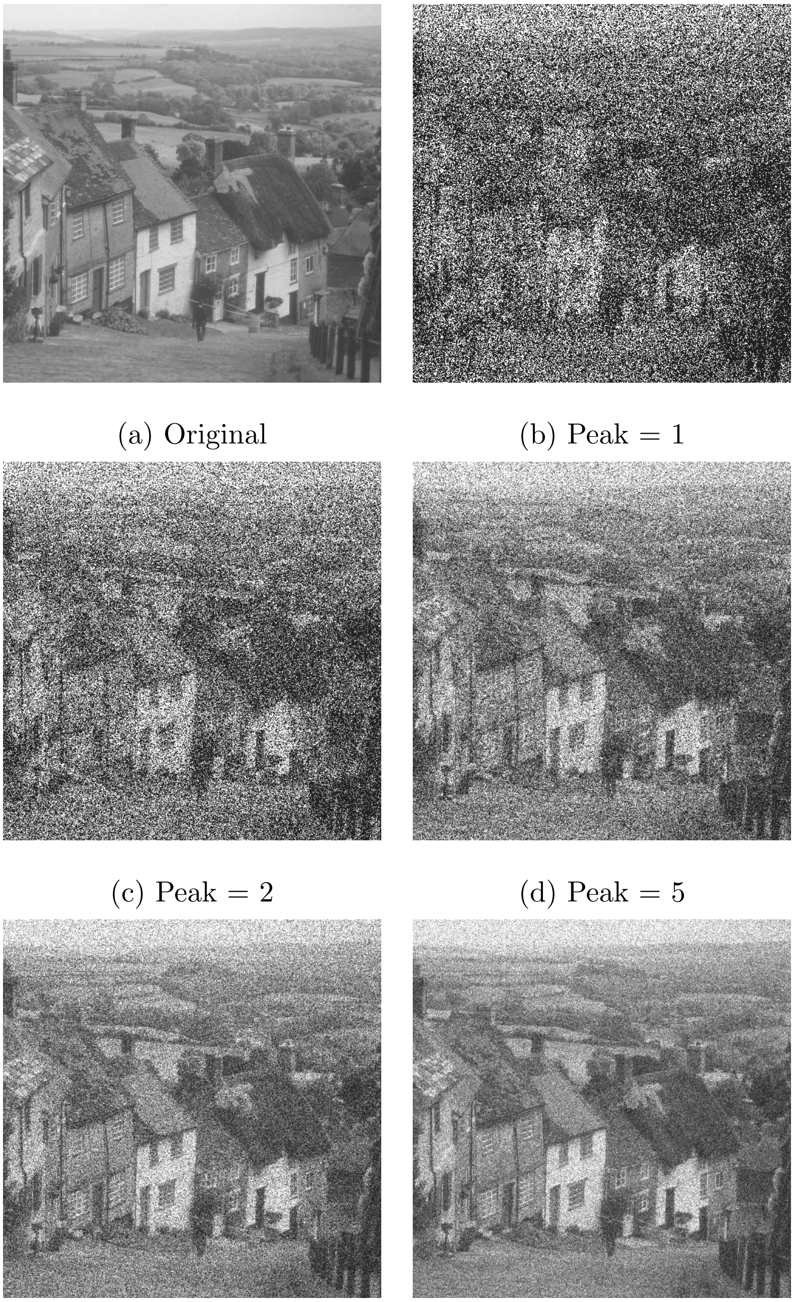
,

In this section previously presented BMCNN framework is extended to non-Gaussian noise cases. First, this dissertation proposes a BMCNN-based image denoising method for Poisson noise, which is another type of artificial noise. Then experiments in real noise cases, which are very practical problems, are presented. The SCN framework is also adopted to the real noise.

5.1 Denoising of Poisson Noise

5.1.1 Poisson Noise

Poisson noise an widely-used noise model along with the Gaussian noise, which shows distinguishable characteristic. Above all, Poisson noise is a data-dependent noise model whereas Gaussian noise is independent to the pixel value. In detail, noisy



(a) Original

(b) Peak = 1

(c) Peak = 2

(d) Peak = 5

(e) Peak = 10

(f) Peak = 20

Figure 5.1: An example of the Poisson noise

observation $Y(i,j)$ with an ground-truth image pixel $X(i,j)$ follows the Poisson distribution :

$$P(Y(i,j)|X(i,j)) = \frac{(X(i,j))^{Y(i,j)}}{(Y(i,j))!} exp(-X(i,j)) \tag{5.1}$$

where $Y(i,j)$ is a non-negative integer. The signal-to-noise ratio of each pixel is $\sqrt{X(i,j)}$ and it implies that lower intensity in the image yields a stronger noise. Hence the power of Poisson noise is generally defined by the maximal value of $X(i,j)$, named *peak value*. Fig. 5.1 shows an example of a Poisson noisy image with several peak values.

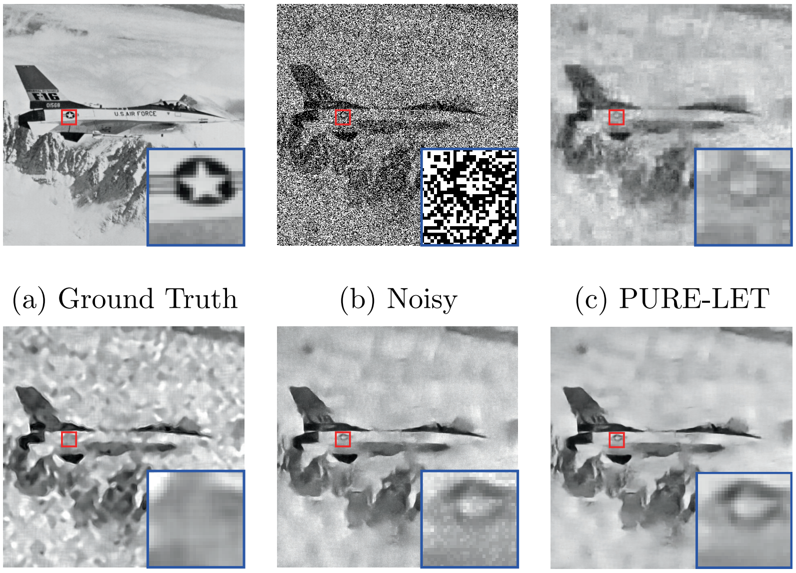
5.1.2 Training Criteria

The BMCNN for Poisson noise is implemented in two-step framework which is same to the BMCNN for Gaussian noise illustrated in Fig. 4.1. : the first step estimates the pilot signal for the block matching, and the second step calculates the final result. There are two main differences. First, since this is the first CNN based approach for Poisson denoising, network for the first step as well as the second step is trained. It also helps analyzing the effect of NSS prior on the Poisson denoising. Second, in this section, a blind denoising network which covers not only a single noise level but a wide range of noise is trained. For this purpose, the training dataset consists of pairs of noisy image and ground truth image with peak values of [1, 2, 5, 10, 20].

The training dataset is constructed from the BSDS set [70], which is cropped to the size of 180 × 180. For the first step, 45 × 45 size patches are extracted with stride 45 and block matching is not applied. The patches are augmented by the FR operation and the total number of the training sample is 256,000. The DB for the second step is constructed by the block-matching on the pilot image similar to the Gaussian noise case. the block size and the stride is set to $20 \times 20 \times 4$ and 20 respectively. The rest parameters including network structure, learning rate are same to that is described in Section. 4.1.

5.1.3 Experiments

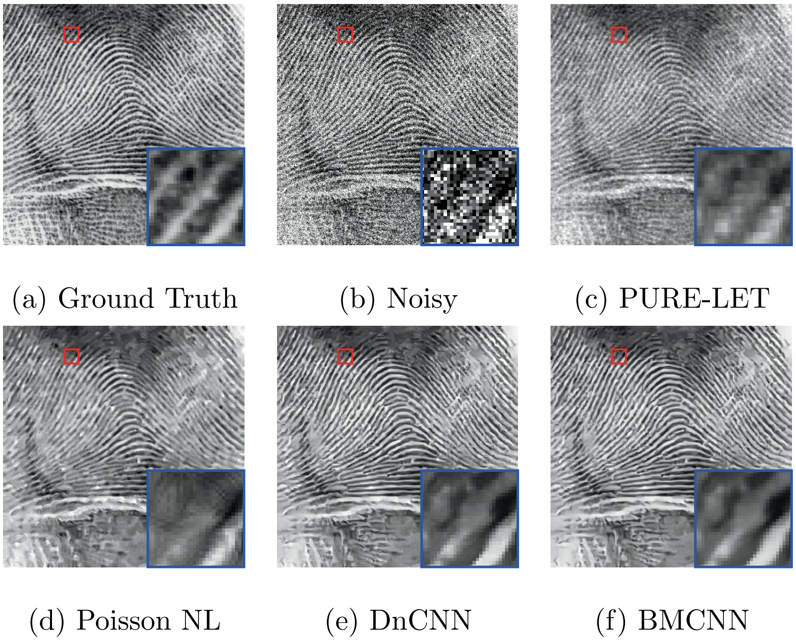
In this subsection, a subjective and objective comparison of the BMCNN method with other state-of-the-art approaches for Poisson denoising - PURE-LET [17] and Poisson nonlocal means filter (PNL) [20] - is presented. The performance of Poisson DnCNN, the first step to construct the pilot signal, is also presented. Among the methods, DnCNN and BMCNN are blind algorithm whereas PURE-LET and PNL are non-blind algorithm, which assumes that the peak value is known and parameters are fixed according the peak value. Table. 5.1 shows the PSNR results of different methods and Figs. 5.2-5.4 illustrates the visual comparisons. As one can see, neural network based DnCNN and BMCNN achieve better PSNR results than conventional prior-based methods, although they are not aware of the noise parameter. However, image prior shows its meaningfulness in two aspects. First, the proposed BMCNN method shows better performance than DnCNN, which is a plain CNN method. Moreover, the PSNR difference between BMCNN and DnCNN is larger in higher peak value, which implies weaker noise. With a very strong noise, even a denoised image is still noisy. For example, the average PSNR of a denoised image, whose original peak value is 1 is about 22dB. The value is similar to the PSNR of a Gaussian-noisy image with $\sigma = 20$. Therefore the block matching is erroneous and the improvement from DnCNN is restricted in low peak value cases. Second, the PNL algorithm shows the best performance in *Barbara*, which is dominated by repetitive texture. In detail, DnCNN is better in low peak values less than 5 and PNL shows outstanding performance in high peak values, where more accurate patch similarity is available. Interestingly, BMCNN shows worse performance in *Barbara* image than DnCNN. The reason is illustrated in Fig. 5.4. Although the block matching tries to make use of the NSS prior and improve the performance in regular structures, most textures are already blurred out in the first step. Therefore, the pilot signal cannot provide the accurate block matching and the resulting BMCNN produces over-smooth textures.



(a) Ground Truth (b) Noisy (c) PURE-LET

(d) Poisson NL (e) DnCNN (f) BMCNN

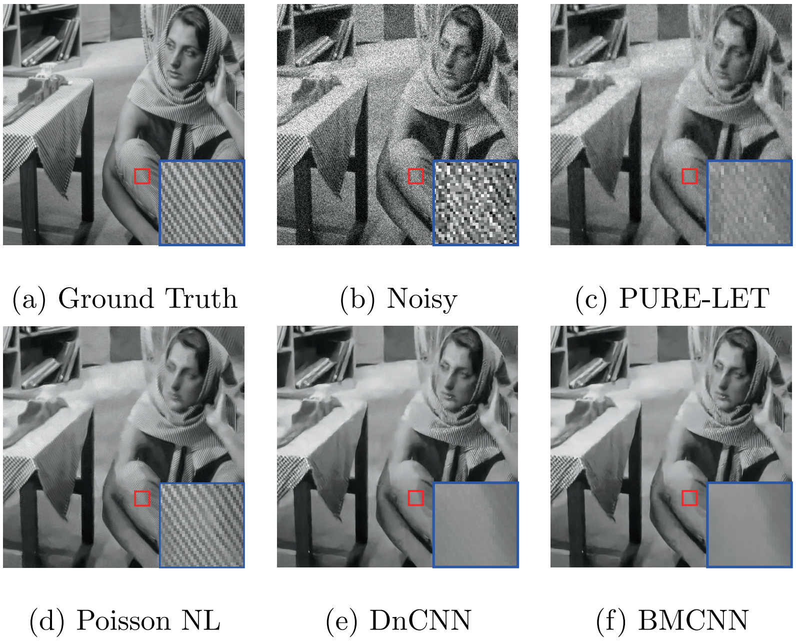
Figure 5.2: Denoising result of the *Jetplane* with peak value = 1.



(a) Ground Truth (b) Noisy (c) PURE-LET

(d) Poisson NL (e) DnCNN (f) BMCNN

Figure 5.3: Denoising result of the *Fingerprint* with peak value = 5.



(a) Ground Truth (b) Noisy (c) PURE-LET

(d) Poisson NL (e) DnCNN (f) BMCNN

Figure 5.4: Denoising result of the *Barbara* with peak value = 50.

Table 5.1: PSNR of different denoising methods. The best results are highlighted in **bold**.

Method	Noisy	PURE-LET	Poisson NL	DnCNN	BMCNN
			Peak value = 1		
Cameraman	6.61	19.82	19.97	**21.71**	21.70
Lena	6.24	22.40	22.58	23.85	**23.93**
Barbara	6.43	20.31	20.56	21.12	**21.16**
Boat	6.17	21.28	20.90	22.02	**22.10**
Couple	6.30	21.27	20.98	22.00	**22.04**
Fingerprint	5.93	17.19	16.31	17.25	**17.31**
Hill	6.45	22.38	22.20	23.11	**23.20**
House	5.98	21.34	21.29	22.76	**22.79**
Jetplane	5.42	21.32	20.63	22.40	**22.62**
Man	6.51	21.74	21.91	22.81	**22.91**
Montage	7.20	19.31	20.00	**21.89**	21.75
Peppers	6.34	19.30	19.78	**20.94**	20.90
Average	6.30	20.64	20.59	21.82	**21.87**

Method	Noisy	PURE-LET	Poisson NL	DnCNN	BMCNN
$Peak\ Value = 2$					
Cameraman	8.79	20.89	21.16	**22.87**	22.77
Lena	8.37	23.49	24.21	25.33	**25.41**
Barbara	8.56	21.21	21.69	22.02	**22.04**
Boat	8.30	22.32	22.18	23.42	**23.44**
Couple	8.38	22.36	22.25	23.20	**23.20**
Fingerprint	8.10	18.29	17.31	18.71	**18.94**
Hill	8.54	23.53	23.60	24.26	**24.31**
House	8.11	22.60	23.36	24.79	**24.88**
Jetplane	7.77	22.53	22.34	24.00	**24.17**
Man	8.59	22.81	23.27	23.97	**24.00**
Montage	9.38	21.18	21.47	23.67	**23.67**
Peppers	8.45	20.39	21.22	22.82	**22.86**
Average	8.45	21.80	22.00	23.25	**23.31**

Method	Noisy	PURE-LET	Poisson NL	DnCNN	BMCNN
$Peak\ Value = 5$					
Cameraman	11.74	22.67	21.08	**25.19**	25.09
Lena	11.35	25.30	26.28	27.42	**27.47**
Barbara	11.61	22.32	**23.31**	23.09	23.03
Boat	11.21	23.75	23.91	25.15	**25.21**
Couple	11.31	23.50	23.76	**24.84**	24.83
Fingerprint	11.13	20.06	19.70	20.98	**21.42**
Hill	11.61	24.81	25.07	25.83	**25.83**
House	11.09	24.05	25.54	27.10	**27.17**
Jetplane	10.92	24.05	24.38	25.94	**26.08**
Man	11.62	24.22	24.71	25.54	**25.57**
Montage	12.49	23.62	24.40	26.05	**26.21**
Peppers	11.48	22.12	22.63	25.13	**25.25**
Average	11.46	23.37	23.73	25.19	**25.27**

$Peak\ Value = 10$

Method	Noisy	PURE-LET	Poisson NL	DnCNN	BMCNN
Cameraman	14.09	24.10	25.09	**26.36**	26.32
Lena	13.82	26.69	27.72	28.77	**28.88**
Barbara	14.11	23.18	**24.95**	23.94	23.86
Boat	13.61	24.71	25.32	26.35	**26.48**
Couple	13.80	24.54	25.07	26.32	**26.32**
Fingerprint	13.58	21.41	21.67	22.50	**23.04**
Hill	14.19	25.71	26.14	**27.05**	27.04
House	13.55	25.75	27.40	28.85	**29.00**
Jetplane	13.33	25.67	25.92	27.26	**27.50**
Man	14.12	25.32	25.86	26.73	**26.76**
Montage	14.05	25.65	26.38	26.70	**28.05**
Peppers	13.94	23.94	25.23	26.74	**26.82**
Average	13.93	24.72	25.56	26.46	**26.67**

Method	Noisy	PURE-LET	Poisson NL	DnCNN	BMCNN
$Peak\ Value = 20$					
Cameraman	16.60	25.64	26.62	27.24	**27.61**
Lena	16.48	27.94	29.27	30.34	**30.38**
Barbara	16.80	24.35	**26.92**	25.05	24.81
Boat	16.19	26.15	26.80	27.78	**27.88**
Couple	16.47	26.05	26.58	**27.89**	27.82
Fingerprint	16.22	22.93	23.60	24.40	**24.83**
Hill	16.94	27.10	25.35	**28.33**	28.29
House	16.17	27.52	29.07	30.30	**30.50**
Jetplane	15.75	27.21	27.61	29.00	**29.14**
Man	16.82	26.58	27.03	**28.04**	28.01
Montage	17.69	26.95	28.05	24.90	**29.24**
Peppers	16.55	26.06	27.16	28.34	**28.43**
Average	16.56	26.21	27.00	27.63	**28.08**

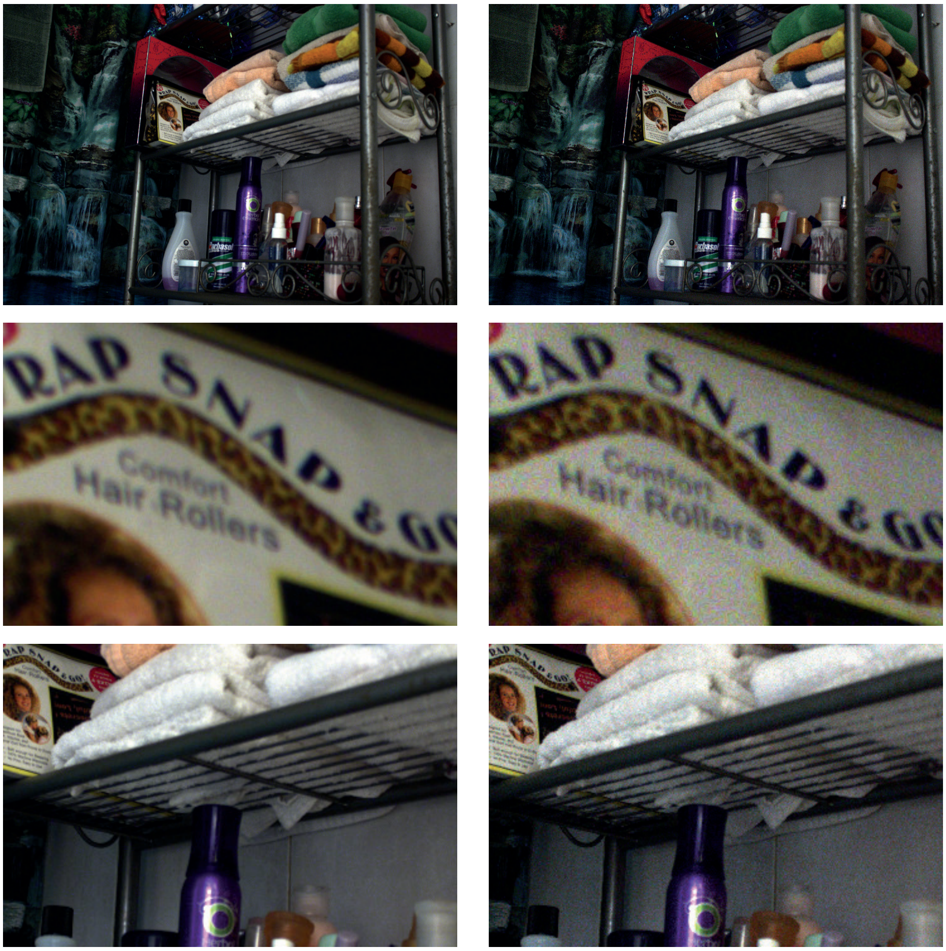
5.2 Denoising of Real Noise

Up-to-date studies of image deonising assume various noise models. Recently, a mixed model also attracts attention since a single model is insufficient to represent the real cases, where an image is disrupted by multiple noise sources. However, a noise from the model is an artificial noise in the end, and it would have different characteristics to the noise that occurs in a real case. From this aspect, this section presents a denoising algorithm for a real noise.

5.2.1 Real Noise Database

In real cases, low-light conditions yield severe degradation in a captured image due to insufficient exposure time and small sensor size. Recently presented RENOIR [41] dataset provides a real low-light image noise dataset from this aspect. The image is composed of pairs of noisy image, which is obtained with high light sensitivity and short exposure time and reference image, which is obtained with low light sensitivity and long exposure time. An image captured with long exposure time tends to be blurry due to object motion or camera shivering. However, it contains much less noise and the blur is negligible when capturing a static scene using a tripod. Therefore, long exposure image is a good approximation for the ground truth image. Fig. 5.5 shows an example of the dataset.

In this dissertation, all the image is resized by a scale factor of 1/6 for the following reasons. First, the image size is about 18 Mpixels in maximum and it is so huge that implementing deep network with the image can cause out of memory problems. Second, although a tripod is used to prevent shivering, there exists a displacement between a reference image and a corresponding noisy image. The displacement is



(a) Reference

(b) Noisy

Figure 5.5: An example of the RENOIR dataset. First row : full images, second row : cropped region of the original images, third row : cropped region of the resized images. The sizes of regions in second row and third row are same.

amount to 4 pixels in horizontal and vertical simultaneously in maximum. Therefore the error between a denoised image and a reference image can be mainly due to the misalignment rather than noise. However, the displacement is one pixel in maximum in down-sampled image and its effect can be compensated. Although the experiments are constructed for the low-resolution image as a result of resizing, it is believed that the denoising algorithms would show similar results for higher-resolution images.

The RENOIR dataset is composed of images that are captured from three different cameras : a compact digital camera (Canon S90), a DSLR (Canon T3i), and a mobile phone camera (Xiaomi Mi3). In this dissertation, a individual BMCNN denoising network is trained and tested for each camera model. The dataset contains 40 scenes per camera. Among them, 35 scenes (Scene 1∼35) are used as a training set and remaining 5 scenes (Scene 36∼40) constitute test set.

5.2.2 Training Criteria

The training criteria for the Real noise is similar to the Poisson noise case described in Subsection. 5.1.2. The only difference is that real images are RGB color images where the images used for the Poisson noise experiments are grayscale images. Since color image denoising needs to analyze the multiple channel simultaneously, it needs more information than grayscale image denoising. Therefore, the training patch for the first step is set to $50 \times 50 \times 3$, which is three channel and spatially larger than the Poisson case. However, increasing the patch size of the second step may not be favorable as shown in 4.2.4. Hence the block size is $20 \times 20 \times 12$, where 4 patches are stacked by the channel direction.

5.2.3 Experiments

The proposed BMCNN algorithm is compared with some state-of-the-art image denoising methods: BM3D [3] and DnCNN [30]. There exist two DnCNN networks in this experiments. The first one is the first step of the BMCNN algorithm, which is trained from the training set (Scene 1∼35) of the real image dataset. The second one is that trained for a Gaussian noise case, which shows if a denoising network for the artificial noise works well for real images. For distinction, the first network is named DnCNN-Real and the second one is named DnCNN-Art. The BMCNN and DnCNNs are trained in a blind procedure. In detail, images with a diversity of noises compose the training set and as a result, the trained network is suitable for a wide range of noises, including a noise with unknown characteristics. Therefore, the networks are applied to the test set directly The BM3D, on the other hands, is a non-blind algorithm which needs a noise level as an input parameter. Hence for the fair comparison, a noise level estimation algorithm [76] is applied as a preprocessing step. The estimated noise variance is provided for the BM3D algorithm.

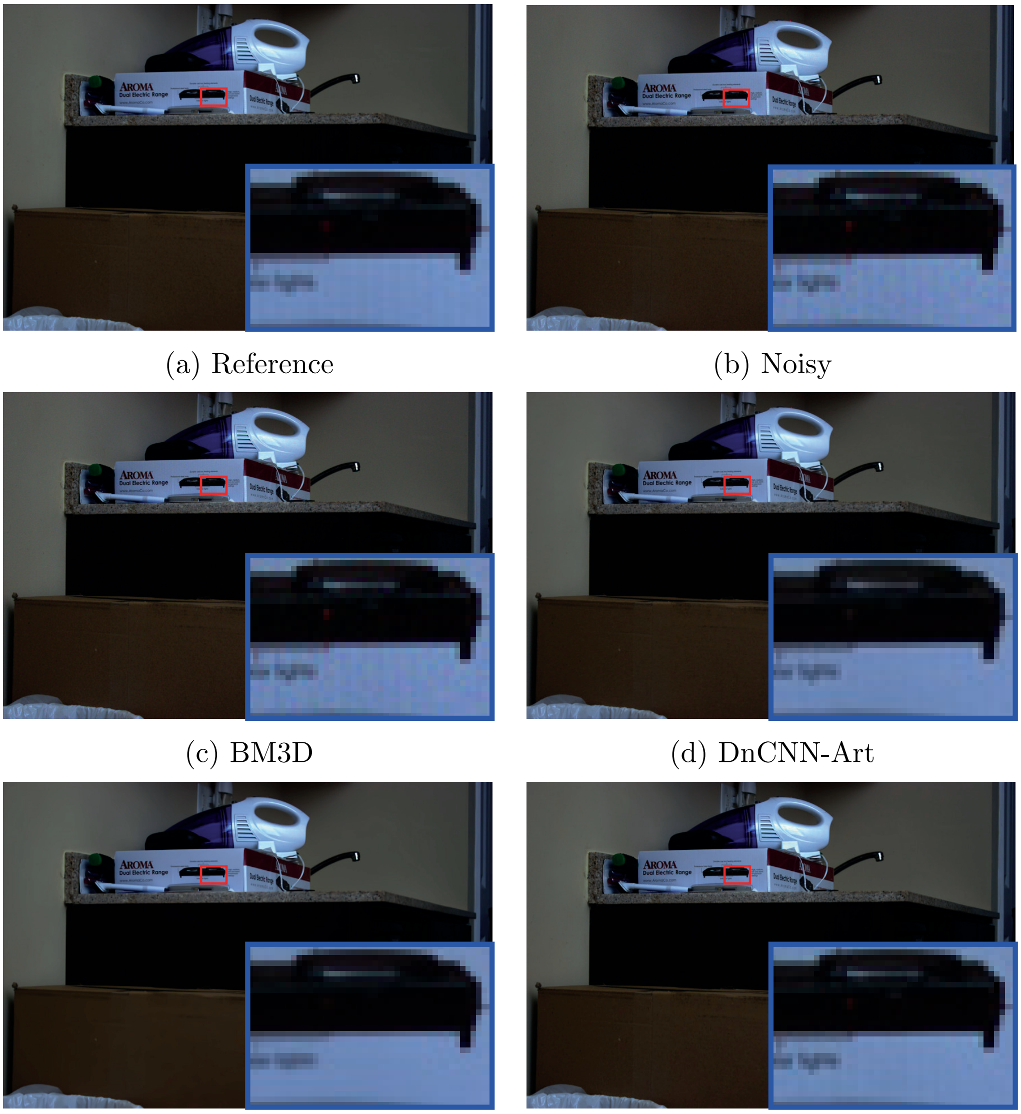
Table. 5.2, 5.3 and 5.4 summarize the average performance and run time of the denoising algorithms for each camera model. In real cases, each camera has distinct model and the noise characteristic is different from time to time. Therefore the denoising results show case-by-case characteristic. However, one can also find a general tendency.

First, since the real noise is comprised of various types of noise, the noise level estimation [76] which estimates the variance of Gaussian noise returns the smaller value than ground-truth error variance. Therefore, the resulting BM3D acts as a weak denoiser which results a small change from the input signal. As a result, BM3D shows its advantages on WSNR and IFC, which place emphasis on the detail conservation.

Second, although DnCNN-Art is trained for the Gaussian noise cases, it can compete with DnCNN-Real, which is trained for each camera model. In detail, DnCNN-Art shows better performance than DnCNN-Real for every metric in S90 dataset whereas DnCNN-Real takes advantages in Mi3 dataset. In Ti3, DnCNN-Art is better in the view of WSNR, SSIM and IFC. It also implies that Gaussian model is quite reasonable model for real noise.

Finally, proposed BMCNN shows the best performance for the most cases. In the view of PSNR, which is the most general metric for image processing, BMCNN is always the best algorithm. Fig. 5.6 - 5.11 illustrate the results for various images and shows how BMCNN upbuilds its first step, DnCNN-Real. Fig. 5.8 and 5.9 present that DnCNN-Real method makes unwanted artifacts on smooth regions in S90 dataset. The reason can be inferred from Fig. 5.12. As shown, S90 dataset contains large amount of rough surfaces where artifact-like textures are present. Hence the DnCNN-Real network that is trained from the S90 dataset tends to reconstruct detailed textures and makes the noisy output. Since the BMCNN groups similar patches, the noise in the individual patch is canceled out and therefore, the network estimates clean output successfully even when DnCNN-Real fails.

Moreover, although the training dataset are downsampled, there still exist small sub-pixel displacements. In order to deal with varying displacement, DnCNN-Real tends to smooth the image. On the other hands, BMCNN can compensate such displacements by combining a number of non-local patches. Hence the BMCNN prevents over-smoothing and it yields better results.



(a) Reference

(b) Noisy

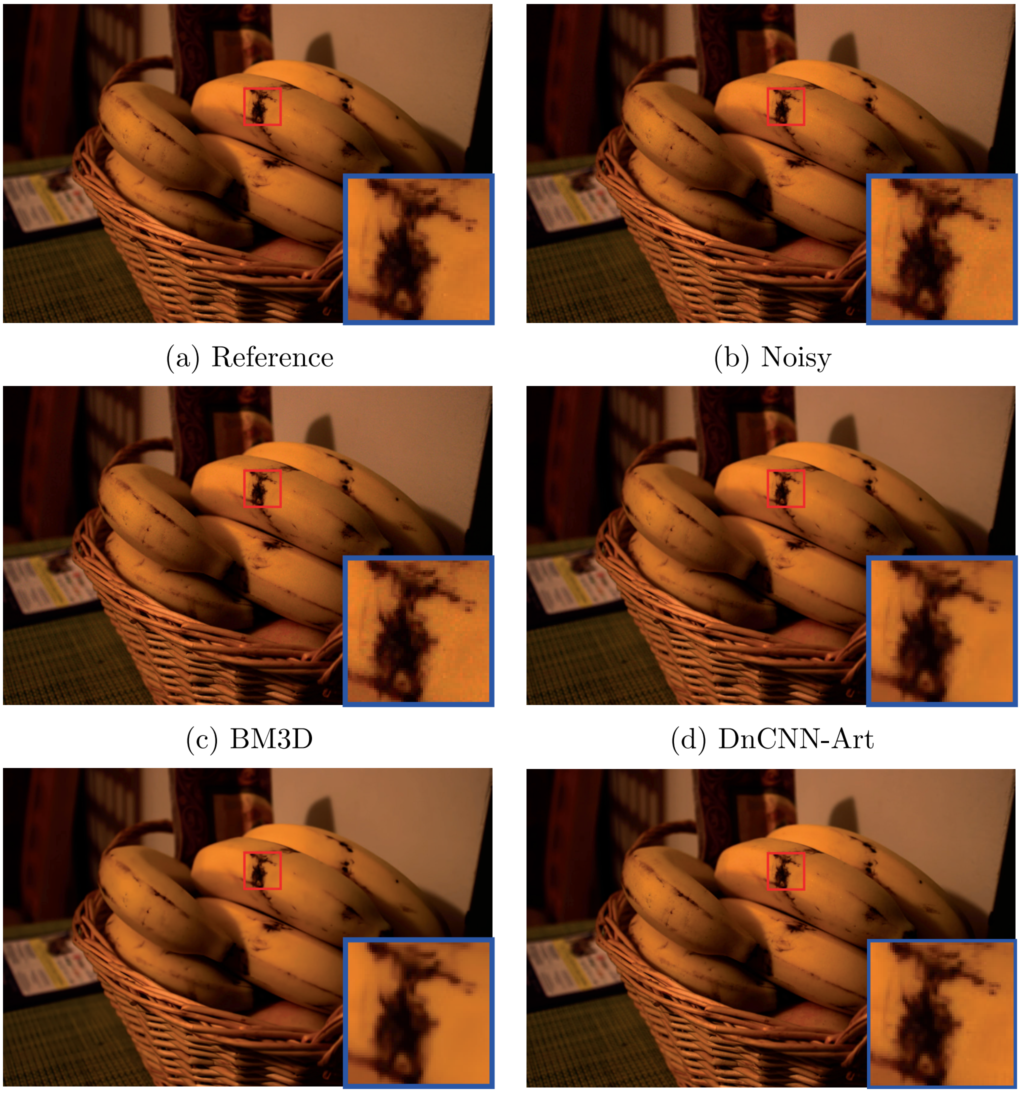
(c) BM3D

(d) DnCNN-Art

(e) DnCNN-Real

(g) BMCNN

Figure 5.6: Denoising results of *scene 36* of T3i dataset.


(a) Reference

(b) Noisy

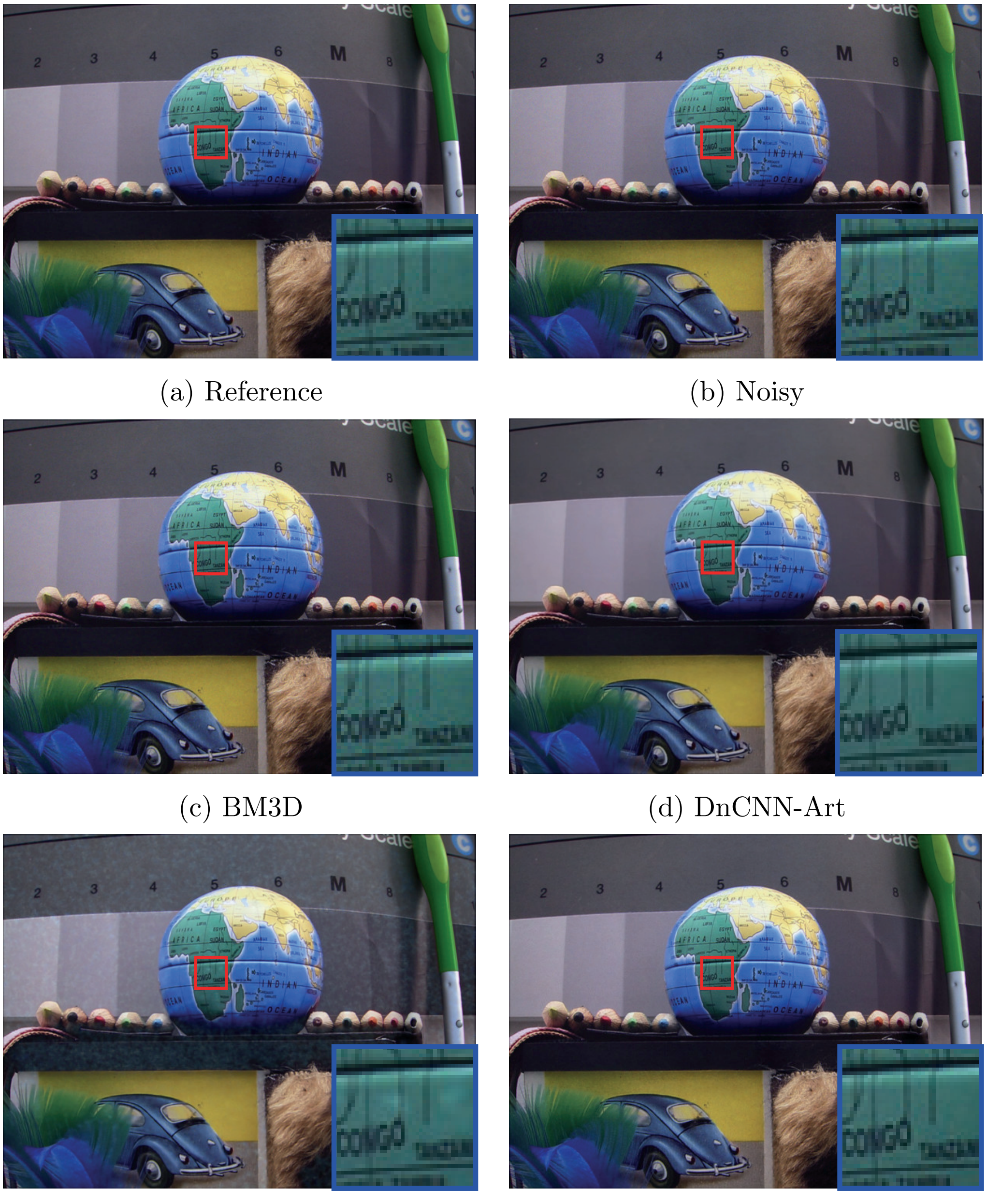
(c) BM3D

(d) DnCNN-Art

(e) DnCNN-Real

(f) BMCNN

Figure 5.7: Denoising results of *scene 40* of T3i dataset.



(a) Reference

(b) Noisy

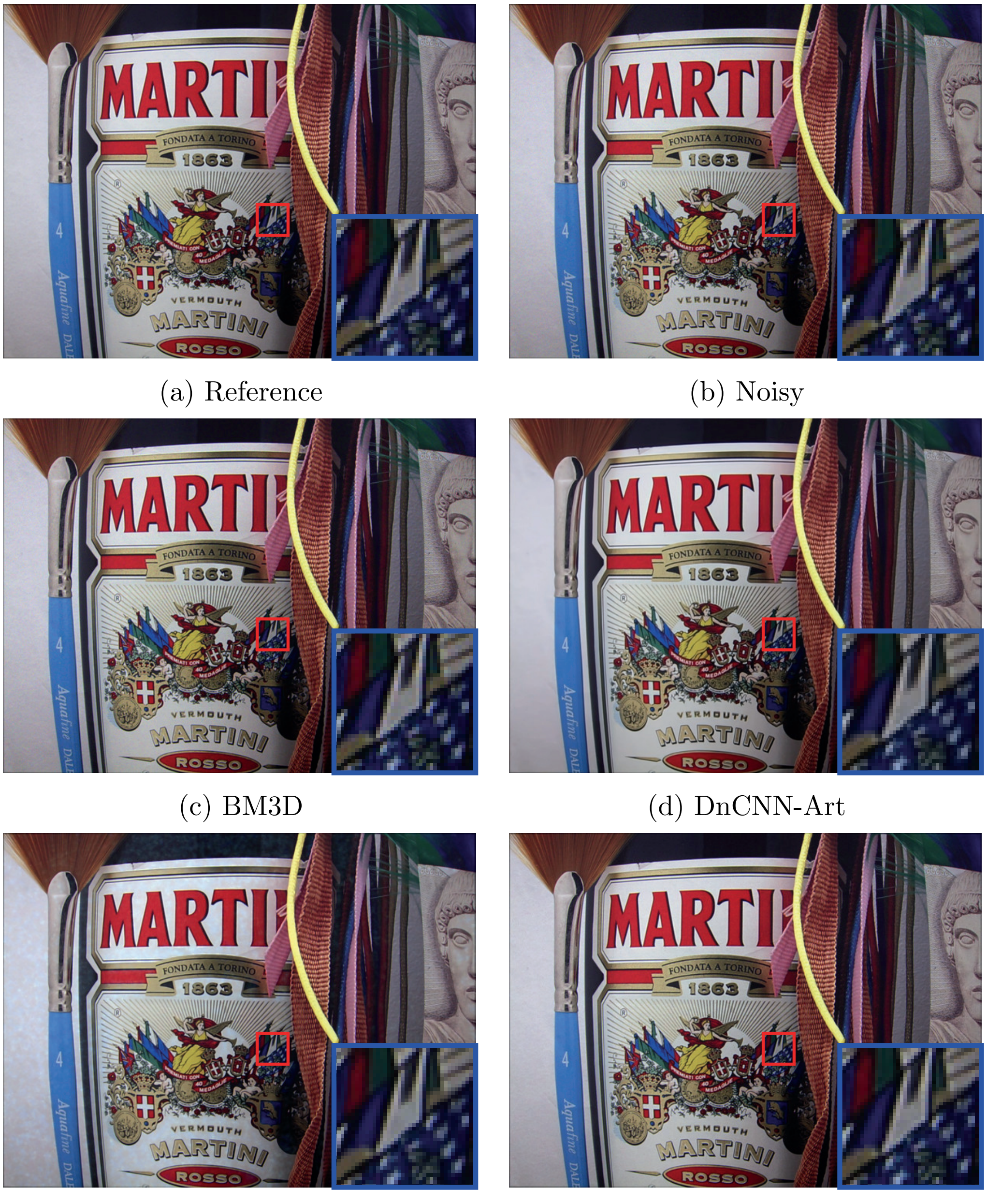
(c) BM3D

(d) DnCNN-Art

(e) DnCNN-Real

(f) BMCNN

Figure 5.8: Denoising results of *scene 39* of S90 dataset.



(a) Reference

(b) Noisy

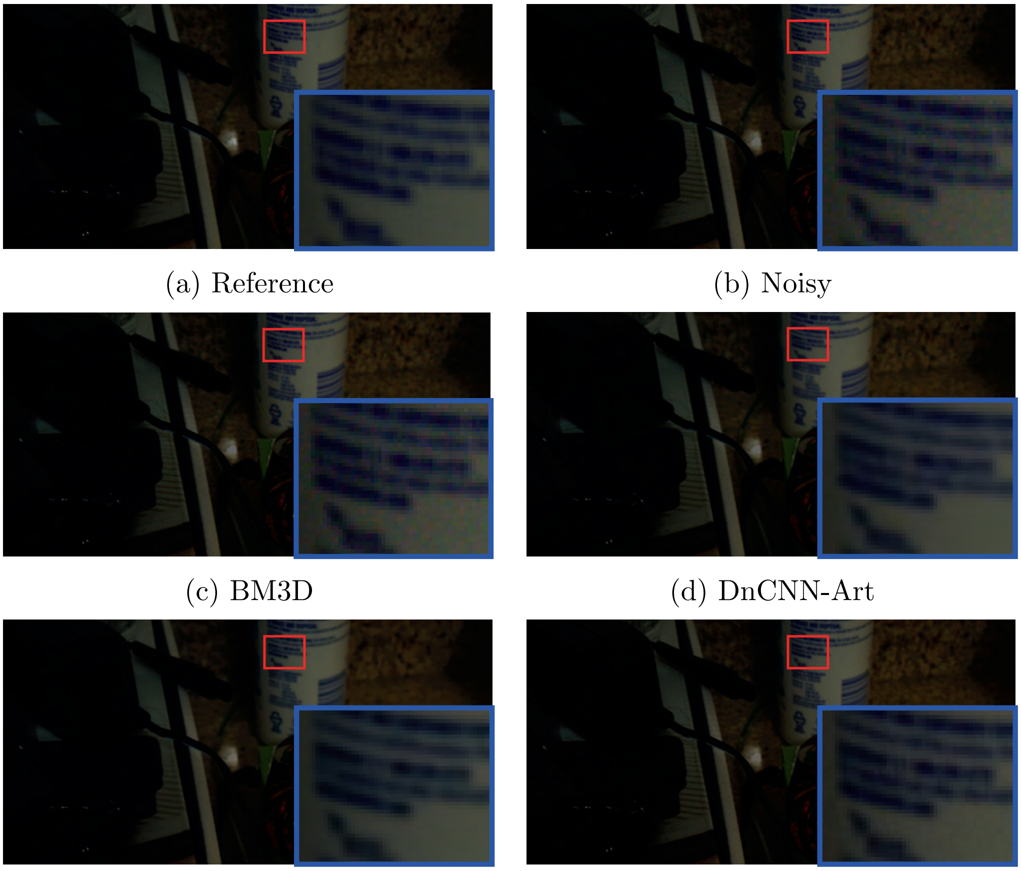
(c) BM3D

(d) DnCNN-Art

(e) DnCNN-Real

(f) BMCNN

Figure 5.9: Denoising results of *scene 40* of S90 dataset.



(a) Reference

(b) Noisy

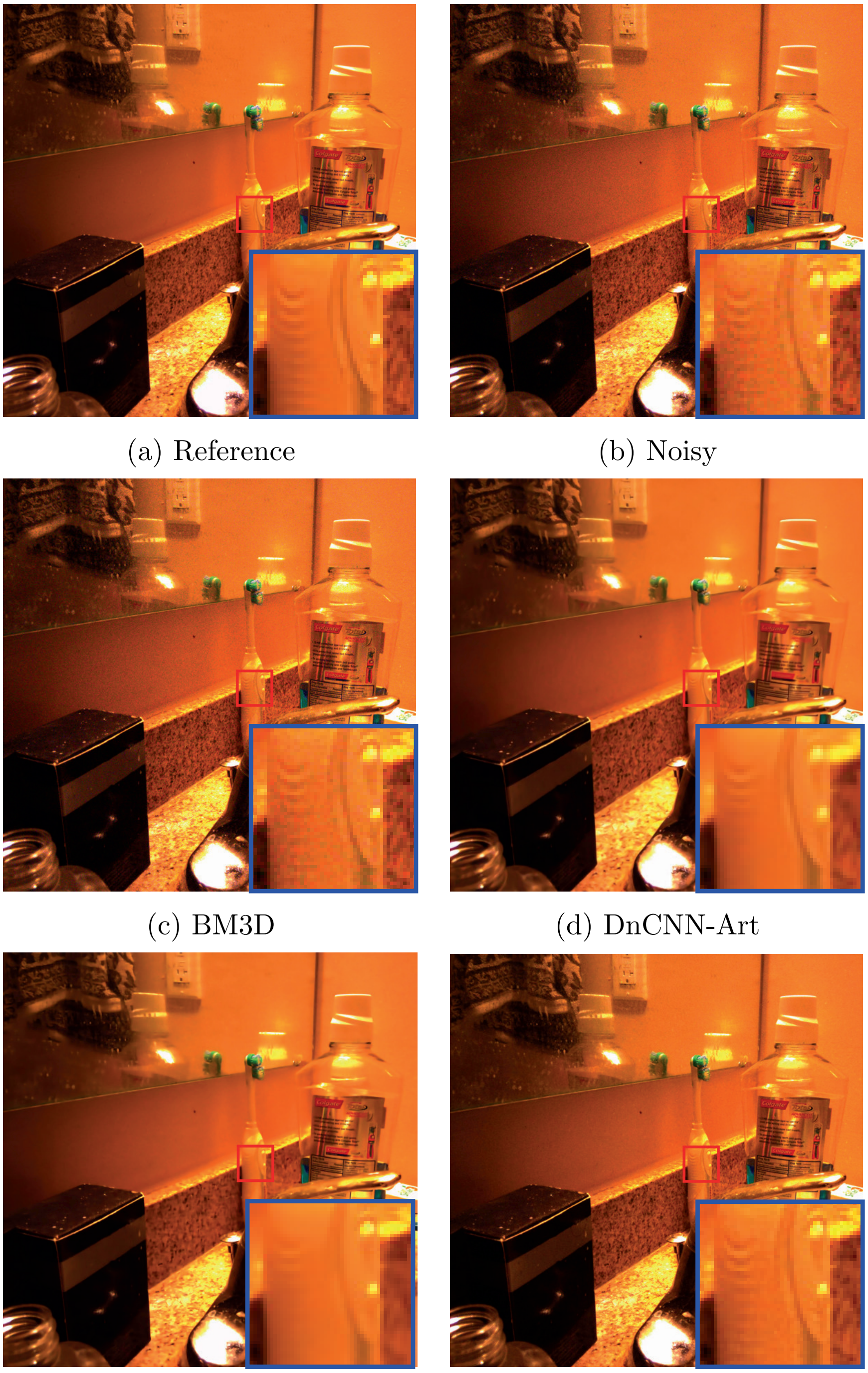
(c) BM3D

(d) DnCNN-Art

(e) DnCNN-Real

(f) BMCNN

Figure 5.10: Denoising results of *scene 36* of Mi3 dataset.



(a) Reference

(b) Noisy

(c) BM3D

(d) DnCNN-Art

(e) DnCNN-Real

(f) BMCNN

Figure 5.11: Denoising results of *scene 38* of Mi3 dataset.

Table 5.2: Average PSNR, WSNR, SSIM, IFC, VIF, and run time of different denoising methods for T3i camera.

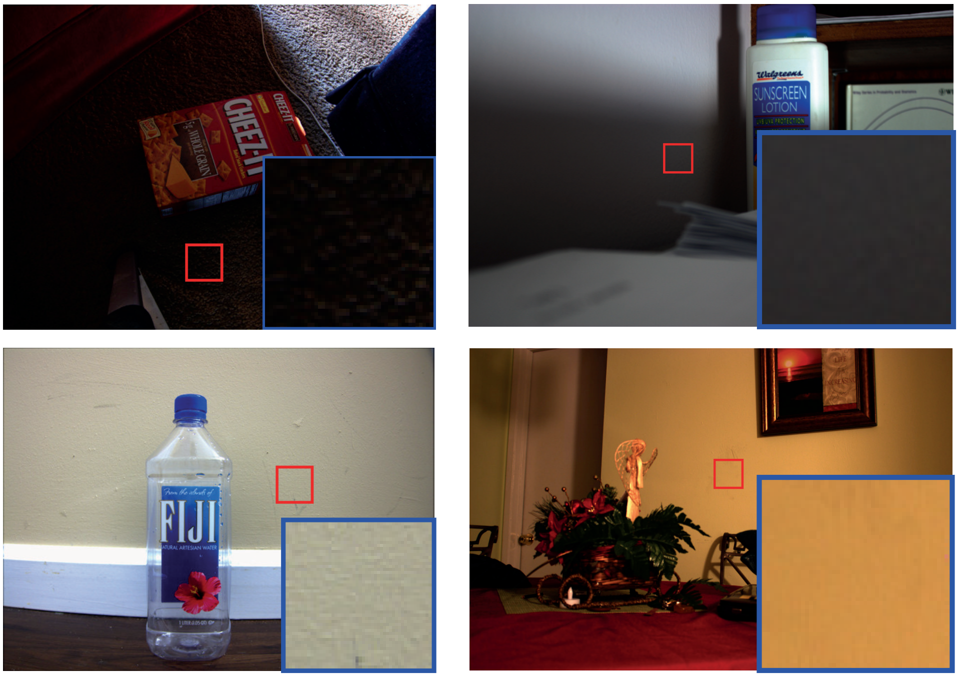
Method	Noisy	BM3D	DnCNN-Art	DnCNN-Real	BMCNN
PSNR	38.98	39.89	40.96	42.10	**43.27**
WSNR	35.96	39.88	39.40	39.39	**40.38**
SSIM	0.8601	0.9875	0.9894	0.9869	**0.9931**
IFC	2.784	**5.028**	4.523	3.648	4.851
VIF	0.4555	0.7904	0.8420	**0.8852**	0.8670
Run Time	-	16.97	0.324	0.819	12.98

Table 5.3: Average PSNR, WSNR, SSIM, IFC, VIF, and run time of different denoising methods for S90 camera.

Method	Noisy	BM3D	DnCNN-Art	DnCNN-Real	BMCNN
PSNR	37.57	38.04	38.53	33.42	**39.01**
WSNR	36.53	**45.04**	44.28	29.05	44.98
SSIM	0.7939	0.9839	**0.9876**	0.9697	0.9866
IFC	3.897	7.871	7.032	6.141	**8.224**
VIF	0.4749	0.8758	**0.9157**	0.8047	0.8866
Run Time	-	8.642	0.157	0.604	8.371

Table 5.4: Average PSNR, WSNR, SSIM, IFC, VIF, and run time of different denoising methods for Mi3 camera.

Method	Noisy	BM3D	DnCNN-Art	DnCNN-Real	BMCNN
PSNR	31.80	31.91	33.10	35.10	**35.88**
WSNR	28.49	31.63	30.60	32.79	**33.24**
SSIM	0.8202	0.9765	0.9682	0.9695	**0.9796**
IFC	3.423	6.332	4.762	4.980	**6.600**
VIF	0.5038	0.8269	0.8355	**0.8913**	0.8420
Run Time	-	6.954	0.107	0.541	8.124



(a) S90

(b) T3i

Figure 5.12: Examples of background surface in $S90$ and $T3i$ dataset.

Effect of SCN

In this subsection, an additional experiment is conducted to show whether the SCN structure can further improve the performance on real images. Every network that is tested on the real camera dataset - DnCNN-Art, DnCNN-Real, BMCNN is employed as a base network to construct a committee - SCN-Art, SCN-Real, SCN-BM respectively. In order to construct a committee, SCN-FR in 3.2 is employed for all experiments.

Table 5.5 shows the PSNR results of the base network and its corresponding committee. In most cases, adopting SCN improves the PSNR. Especially, BMCNN and DnCNN-Real experience drastic PSNR gain by SCN method in T3i and S90 dataset respectively. Fig. 5.13 illustrates how the SCN method improves the performance of BMCNN on T3i dataset. T3i, which is a DSLR camera, is easy to be shaken when its shutter is pressed compared to the other cameras which uses electronic shutters. Therefore, scenes in T3i training set often suffer from misalignment between the noisy image and the reference image. Although the BMCNN compensates the misalignment and the result can preserve details, it often yields unwanted displacement. By SCN framework, that displacement is canceled out while maintain detail. As mentioned above, S90 training set contains artifact-like textures and the trained DnCNN is prone to reconstruct unwanted artifact on the smooth regions. Fig. 5.14 illustrates that the artifacts are canceled out by using the SCN framework.

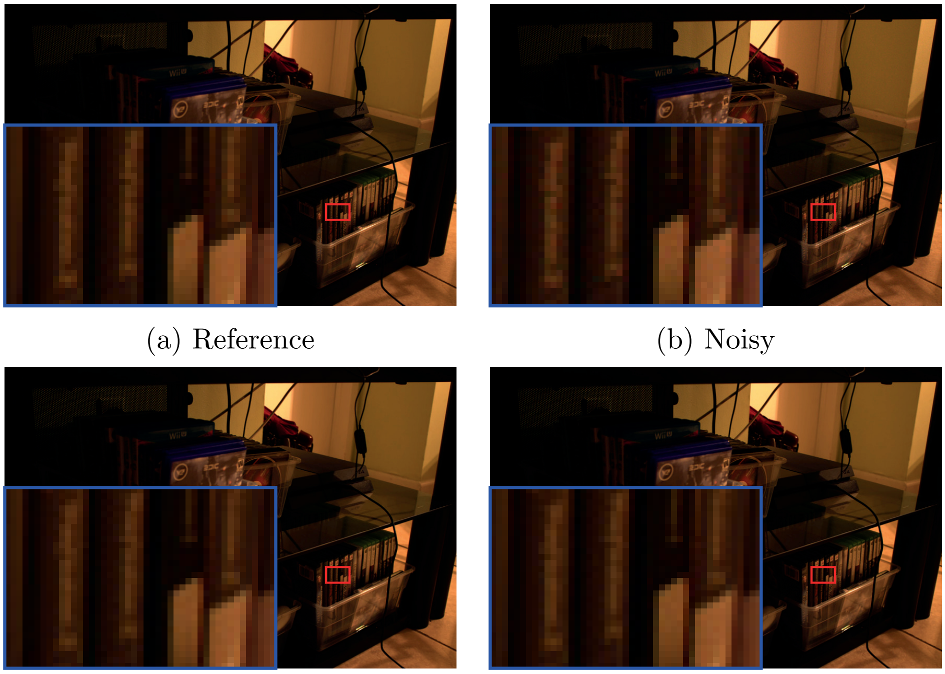
As shown, the SCN method improves the performance of the denoising network in the real cases. In detail, SCN can compensate the difficulties of the networks that arise from the incompleteness of the training dataset. It yields larger PSNR gain over the base network compared to the Gaussian noise denoising cases. From this experiment, the SCN is expected to be an effective structure when treating a difficult task, where it is hard to train an optimal neural network.

Table 5.5: Average PSNR, WSNR, SSIM, IFC, and VIF of different denoising methods for various cameras.

T3i	DnCNN-Art	SCN-Art	DnCNN-Real	SCN-Real	BMCNN	SCN-BM
PSNR	40.96	41.08	42.10	42.33	43.27	**43.92**
WSNR	39.40	39.55	39.39	39.55	40.38	**40.70**
SSIM	0.9894	0.9897	0.9869	0.9872	0.9931	**0.9937**
IFC	4.523	4.628	3.648	3.736	4.851	**4.971**
VIF	0.8420	0.8504	0.8852	**0.8956**	0.8670	0.8836

T3i	DnCNN-Art	SCN-Art	DnCNN-Real	SCN-Real	BMCNN	SCN-BM
PSNR	38.53	38.67	33.42	33.75	39.01	**39.08**
WSNR	44.28	44.56	29.05	29.44	44.98	**45.02**
SSIM	0.9876	**0.9881**	0.9697	0.9721	0.9866	0.9867
IFC	7.032	7.203	6.141	6.308	8.224	**8.247**
VIF	0.9157	0.9243	0.8047	0.8215	0.8866	0.8878

T3i	DnCNN-Art	SCN-Art	DnCNN-Real	SCN-Real	BMCNN	SCN-BM
PSNR	33.10	33.16	35.10	35.27	35.88	**35.90**
WSNR	30.60	30.73	32.79	33.01	33.24	**33.29**
SSIM	0.9682	0.9687	0.9695	0.9704	0.9796	**0.9798**
IFC	4.762	4.857	4.980	5.128	6.600	**6.614**
VIF	0.8355	0.8447	0.8913	**0.9063**	0.8420	0.8429



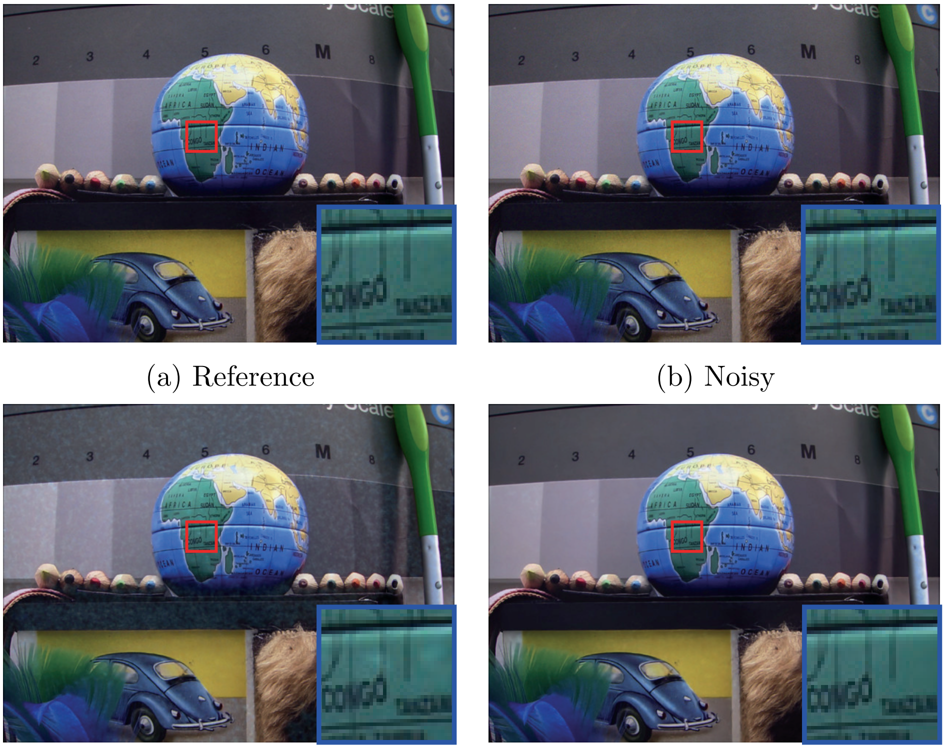
(a) Reference

(b) Noisy

(c) BMCNN

(d) SCN-BM

Figure 5.13: An illustration of the effect of the SCN on T3i-38 scene



(a) Reference

(b) Noisy

(c) DnCNN-Real

(d) SCN-Real

Figure 5.14: An illustration of the effect of the SCN on S90-39 scene

Chapter 6

Conclusions

In this dissertation, algorithms for the image denoising based on the convolutional neural network have been proposed. The proposed algorithms employs the prior models to the existing network structures.

First, a self-committee method to improve the performance of CNN based image restoration algorithms is presented. Unlike the existing approaches that use several differently trained networks as the committee members, a single network is employed and the outputs of transformed inputs are used as the committee member. The transforms used to construct the committee are based on the image processing priors. The transfomed inputs induce different feature maps from the original, and thus produces the outputs with different characteristics. Hence averaging the outputs from differently transformed inputs could enhance the restoration performances. Experiments show that the proposed method enhances the performance of state-of-the-art image denoising and SISR networks.

A framework that combines two dominant approaches in up-to-date image denoising algorithms, i.e., the NSS prior based methods and CNN based methods, is also proposed. Specifically, a CNN that estimates the noise component from a group of nonlocal similar patches is trained. Unlike the conventional NSS based methods, the proposed denoiser is trained with vast data to learn the optimal mapping function and thus achieves better performance. The BMCNN also shows better performance than the existing CNN based method especially in the case of images with regular structure, because the BMCNN considers NSS in addition to the local characteristics.

The proposed algorithms are extended to the non-Gaussian noises: Poisson noise and real noise. Experiments have shown that the BMCNN shows the best performance regardless of the noise type. Especially, conventional CNN based methods can fail on the real noise case where the ground-truth image is unavailable and the reference image, which plays a role of a label, is noisy and shows displacement with the input noisy image. Block-matching step and the SCN structure assist the CNN network to compensate these problems and achieve robust performance.

Bibliography

[1] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 60–65.

[2] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with bm3d?” in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2392–2399.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Trans. Image process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Trans. Image process.*, vol. 22, no. 4, pp. 1620–1630, 2013.

[5] W. Dong, G. Shi, and X. Li, “Nonlocal image restoration with bilateral variance estimation: a low-rank approach,” *IEEE Trans. Image process.*, vol. 22, no. 2, pp. 700–711, 2013.

[6] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image process.*, vol. 16, no. 5, pp. 1395–1411, 2007.

[7] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2862–2869.

[8] O. G. Guleryuz, "Weighted overcomplete denoising," in *Proc. of Asilomar Conference on Signals, Systems and Computer*, 2003, pp. 1992–1996.

[9] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2009, pp. 769–776.

[10] X. Jia, X. Feng, and W. Wang, "Adaptive regularizer learning for low rank approximation with application to image denoising," in *Proc. of International Conference on Image Processing (ICIP)*, 2016, pp. 3096–3100.

[11] B. Wang, T. Lu, and Z. Xiong, "Adaptive boosting for image denoising: Beyond low-rank representation and sparse coding," in *Proc. of the International Conference on Pattern Recognition (ICPR)*, 2016.

[12] S. J. You and N. I. Cho, "A new image denoising method based on the wavelet domain nonlocal means filtering," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1141–1144.

[13] B. Jin, S. J. You, and N. I. Cho, "Bilateral image denoising in the laplacian subbands," *EURASIP Journal on Image and Video Processing*, vol. 2015, no. 1, pp. 1–12, 2015.

[14] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[15] S. Roth and M. J. Black, “Fields of experts: A framework for learning image priors,” in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, vol. 2, 2005, pp. 860–867.

[16] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[17] F. Luisier, C. Vonesch, T. Blu, and M. Unser, “Fast interscale wavelet denoising of poisson-corrupted images,” *Signal Processing*, vol. 90, no. 2, pp. 415–427, 2010.

[18] F. Luisier, T. Blu, and M. Unser, “Undecimated haar thresholding for poisson intensity estimation,” in *Proc. of International Conference on Image Processing (ICIP)*, 2010, pp. 1697–1700.

[19] ——, “Image denoising in mixed poisson–gaussian noise,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 696–708, 2011.

[20] C.-A. Deledalle, F. Tupin, and L. Denis, “Poisson nl means: Unsupervised non local means for poisson noise,” in *Proc. of International Conference on Image Processing (ICIP)*, 2010, pp. 801–804.

[21] Q. Jin, I. Grama, and Q. Liu, “A new poisson noise filter based on weights optimization,” *Journal of Scientific Computing*, vol. 58, no. 3, pp. 548–573, 2014.

[22] J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett, “Poisson noise reduction with non-local pca,” *Journal of Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 279–294, 2014.

[23] B. Zhang, M. Fadili, J.-L. Starck, and J.-C. Olivo-Marin, “Multiscale variance-stabilizing transform for mixed-poisson-gaussian processes and its applications in bioimaging,” in *Proc. of International Conference on Image Processing (ICIP)*, vol. 6, 2007, p. 233.

[24] A. Jezierska, E. Chouzenoux, J.-C. Pesquet, and H. Talbot, “A primal-dual proximal splitting approach for restoring data corrupted with poisson-gaussian noise,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 1085–1088.

[25] A. Jezierska, C. Chaux, J.-C. Pesquet, H. Talbot, and G. Engler, “An em approach for time-variant poisson-gaussian model parameter estimation,” *IEEE Transactions on Signal Processing*, vol. 62, no. 1, pp. 17–30, 2014.

[26] A. Jezierska, J.-C. Pesquet, H. Talbot, and C. Chaux, “Iterative poisson-gaussian noise parametric estimation for blind image denoising,” in *Proc. of International Conference on Image Processing (ICIP)*, 2014, pp. 2819–2823.

[27] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical poissonian-gaussian noise modeling and fitting for single-image raw-data,” *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, 2008.

[28] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.

[29] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[30] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on image Processing*, 2017.

[31] H. C. Burger, C. Schuler, and S. Harmeling, "Learning how to combine internal and external denoising methods," in *German Conference on Pattern Recognition*, 2013, pp. 121–130.

[32] A. A. Bindilatti and N. D. Mascarenhas, "A nonlocal poisson denoising algorithm based on stochastic distances," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1010–1013, 2013.

[33] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2012, pp. 3642–3649.

[34] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *IEEE Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1135–1139.

[35] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.

[36] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, “An iterative regularization method for total variation-based image restoration,” *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[37] Y. Weiss and W. T. Freeman, “What makes a good model of natural images?” in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2007, pp. 1–8.

[38] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black, “Efficient belief propagation with learned higher-order markov random fields,” in *European Conference on Computer Vision(ECCV)*, 2006, pp. 269–282.

[39] S. Z. Li, *Markov random field modeling in image analysis.* Springer Science & Business Media, 2009.

[40] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *Proc. of the International Conference on Computer Vision (ICCV)*, 2009, pp. 2272–2279.

[41] J. Anaya and A. Barbu, “Renoir-a dataset for real low-light image noise reduction,” *arXiv preprint arXiv:1409.8230*, 2014.

[42] N. Remenyi, O. Nicolis, G. Nason, and B. Vidakovic, “Image denoising with 2d scale-mixing complex wavelet transforms,” *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5165–5174, 2014.

[43] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[45] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2014, pp. 2774–2781.

[46] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1067–1074.

[47] S. Geman and C. Graffigne, "Markov random field image models and their applications to computer vision," in *Proc. of the International Congress of Mathematicians*, 1986, p. 2.

[48] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. of European Conference on Computer Vision (ECCV)*, 2014, pp. 184–199.

[49] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image process.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[50] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. of the International Conference on Computer Vision (ICCV)*, 2015, pp. 576–584.

[51] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016.

[52] ——, "Deeply-recursive convolutional network for image super-resolution," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016.

[53] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. of International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[56] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. of Asian Conference on Computer Vision(ACCV)*, 2014, pp. 111–126.

[57] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Computational Imaging*, vol. 3, pp. 47–57, 2016.

[58] J. Li, F. Luisier, and T. Blu, "Deconvolution of poissonian images with the pure-let approach," in *Proc. of International Conference on Image Processing (ICIP)*, 2016, pp. 2708–2712.

[59] S. Haykin and N. Network, *Neural Networks : A comprehensive foundation.* Prentice-Hall, 1999.

[60] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[61] R. C. Gonzalez, "Digital image processing," 2008.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[63] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems(NIPS)*, 2016, pp. 901–901.

[64] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. of the International Conference on Computer Vision(ICCV)*, 2015, pp. 1520–1528.

[65] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[66] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.

[67] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature em-

bedding," in *Proc. of the ACM international conference on Multimedia (ACM MM)*, 2014, pp. 675–678.

[68] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations*, 2015.

[69] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[70] P. Arbelaez, C. Fowlkes, and D. Martin, "The berkeley segmentation dataset and benchmark," *http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds*, 2007.

[71] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[72] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, "Image quality assessment based on a degradation model," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 636–650, 2000.

[73] H. R. Sheikh, A. C. Bovik, and G. De Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2117–2128, 2005.

[74] H. R. Sheikh and A. C. Bovik, "A visual information fidelity approach to video quality assessment," in *The First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2005, pp. 23–25.

[75] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” in *Advances in Neural Information Processing Systems(NIPS)*, 2016, pp. 2802–2810.

[76] X. Liu, M. Tanaka, and M. Okutomi, “Single-image noise level estimation for blind denoising,” *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, 2013.

초록

디지털 영상 취득 과정에서 다양한 잡음이 발생하고 그로 인해 영상의 화질이 저하된다. 영상 잡음 제거는 이러한 잡음을 보정하는 중요한 과정으로, 따라서 다양한 연구가 이루어지고 있다. 잡음 제거는 대표적인 불량조건문제이기 때문에 고전적인 연구는 영상 프라이어 (prior) 에 관한 다양한 모델을 이용하는 잡음 제거 방법을 개발해 왔다. 이러한 알고리즘들에서 결과 영상은 개발자가 설계한 영상의 특성을 따르는 방향으로 복원된다. 최근에는 학습 기반의 영상 잡음 제거 알고리즘도 많이 개발되었다. 이 방법들은 데이터 주도 (data-driven) 방식을 통해 잡음 영상으로부터 복원 영상을 얻어내는 함수를 학습하는 방식으로, 현재 여러 잡음 제거 알고리즘 중 가장 뛰어난 성능을 보이고 있다. 본 논문에서는 컨벌루션 신경망 (CNN)을 기반으로 한 잡음 제거 알고리즘을 제안한다. 기존의 학습 기반 방법들은 픽셀 오차를 최소화하는 데에만 집중하면서 영상의 특성은 고려하지 못하는 데 반해, 제안하는 알고리즘은 영상 프라이어를 고려하여 보다 개선된 결과를 얻도록 한다. 첫 번째로, 기존의 CNN을 여러 개의 변형된 입력 영상에 적용하여 보다 향상된 복원 영상을 얻을 수 있도록 하는 자가위원회 네트워크 방법을 제안한다. 일반적인 가역변환을 겪은 영상을 입력으로 받을 때, CNN 이 만드는 복원 함수는 다른 특성을 갖고, 그에 따라 원본 영상을 입력으로 받는 것과 유사하지만 다른 결과를 생성하게 된다. 기존의 커미트 머신(Committee machine) 방법들로부터 알려져 있듯이 이 결과 영상들을 종합하여 보다 개선된 복원 영상을 얻을 수 있다. 기존의 위원회 기반 방법들은 여러 개의 다른 네트워크를 사용하는 반면, 제안하는 방법은

하나의 네트워크로부터 위원회를 구성한다. 위원회 구성에 사용하는 가역변환은 영상 처리의 특성으로부터 결정된다. 제안하는 자가위원회 네트워크는 별도의 학습 없이도 기존 CNN의 성능을 향상시킨다.

두 번째로, 영상 프라이어를 네트워크 학습에 적용하는 방법을 제안한다. 다양한 프라이어 중 본 논문에서는 잡음 제거에 가장 널리 사용되는 비국부적 자가유사성 (NSS) 프라이어를 적용한다. 제안하는 BMCNN 방법은 블록 매칭을 통해 유사한 패치들을 결합하여 하나의 원본 패치를 예측한다. BMCNN 방법은 다양한 특성을 가진 영상들에서 기존의 방법들보다 뛰어난 성능을 보였다. 또한, 기존의 NSS 기반 방법들은 BMCNN의 구조로 해석될 수 있다.

세 번째로, 가우시안 잡음 이외의 잡음 모델에 대해서도 제안하는 방법들을 적용시킨다. 본 논문에서는 푸아송 잡음과 실제 카메라로부터 촬영한 영상의 Real 잡음에 대한 실험을 구성한다. 제안하는 방법은 다양한 잡음 모델에 대해 좋은 성능을 보였다. 잡음의 특성이 명확하지 않고 영상 좌표의 어긋남이 존재하는 Real 잡음의 경우 특히 뛰어난 성능을 보였다.

주요어: 영상 잡음 제거, 컨벌루션 신경망, 영상 특성, 커미트 머신, 비국부적 자가유사성, Real 잡음

학번: 2011-20871