

Deep Conjunct Denoising and Demosaicking a Hybrid
approach based on Deep Adaptive Residual Learning



Author

Asim Wadood

Regn Number

01-243171-003

Supervisor

Dr. Awais Ahmad

DEPARTMENT

DEPARTMENT OF COMPUTER SCIENCES

BAHRIA UNIVERSITY

ISLAMABAD CAMPUS

FEBRUARY 2019

Deep Conjunct Denoising and Demosaicking a Hybrid approach
based on Deep Adaptive Residual Learning

Author

Asim Wadood

Regn Number

01-243171-003

A thesis submitted in partial fulfillment of the requirements for the degree of
MSCOMPUTER SCIENCE

Thesis Supervisor:

Dr. Awais Ahmad

Thesis Supervisor's Signature:

DEPARTMENT

DEPARTMENT OF COMPUTER SCIENCES

BAHRIA UNIVERSITY

ISLAMABAD CAMPUS

FEBRUARY2018

Declaration

I certify that this research work titled “*Deep Conjunct Denoising and Demosaicking a Hybrid approach based on Deep Adaptive Residual Learning*” is my own work. The work was not presented for evaluation elsewhere. The material used from other sources has been properly recognized / referred to.

Signature of Student

Asim Wadood

01-243171-003



Bahria University
Discovering Knowledge

MS-13

Thesis Completion Certificate

Scholar's Name: Asim Wadood Registration No. 49849

Programme of Study: MS (CS)

Thesis Title: Deep Conjunct Denoising and Demosaicking a Hybrid approach based on Deep Adaptive Residual Learning

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for Evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 15% that is within the permissible limit set by the HEC for the MS/MPhil degree thesis. I have also found the thesis in a format recognized by the BU for the MS/MPhil thesis.

Principal Supervisor's Signature: _____

Date: 14th May 2019 **Name:** Dr. Awais Ahmad



Bahria University
Discovering Knowledge

MS-14A

Author's Declaration

I, Asim Wadood hereby state that my MS thesis titled "Deep Conjunct Denoising and Demosaicking a Hybrid approach based on Deep Adaptive Residual Learning" is my own work and has not been submitted previously by me for taking any degree from this university.

Bahria University

or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my Graduate the university has the right to withdraw/cancel my MS degree.

Name of scholar: Asim Wadood

Date: 14th May 2019



Bahria University
Discovering Knowledge

MS-14B

Plagiarism Undertaking

I, solemnly declare that research work presented in the thesis titled "Deep Conjunct Denoising and Demosaicking a Hybrid approach based on Deep Adaptive Residual Learning" is solely my research work with no significant contribution from any other person. Small contribution / help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and Bahria University towards plagiarism. Therefore, I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the university reserves the right to withdraw / revoke my MS degree and that HEC and the University has the right to publish my name on the HEC / University website on which names of students are placed who submitted plagiarized thesis.

Student / Author's Sign: _____

Name of the Student: Asim Wadood

Acknowledgements

I am grateful to my Creator Allah Subhana-Watala for guiding me at every step of this work and for every new thought that Your scheme in my mind will improve it. Indeed, without Your invaluable help and guidance, I could have done nothing. Whoever helped me throughout my thesis, whether my parents or anyone else was your will, no one is worthy of praise but you.

I am grateful to my beloved parents who raised me when I was unable to walk and continued to support me throughout my life.

I would also like to thank my supervisor Dr. Awais Ahmed in particular for his help during my thesis.

Finally, I would like to express my gratitude to all the people who have given my study valuable help.

*Dedicated to my exceptional parents and adorable siblings, whose
tremendous support and cooperation led me to this great
achievement.*

Abstract

In digital photography pipelines, Denoising and Demosaicking are the most essential key stages. In literature, Convolutional neural networks-based Image demosaicking methods have exhibited tremendous achievement. Nonetheless, as most systems are not sufficiently profound, there is still enough space for the enhancement in performance plus a main challenge that remains to be addressed is to guarantee the visual quality of reconstructed images particularly in the presence of noise corruption with efficient computation. Regardless of huge progress made in the previous decade. For these challenges and motivated by new advances in deep residual networks, this thesis introduces a new Demosaicking and denoising conjunct strategy named MARN-JDD using deep adaptive residual learning on that framework train on an enormous bulk of images, in place of adopting custom adapt filters. Conceptually the proposed framework has two stages, in the first stage residual mosaic and noise image is generated through joint through deep adaptive residual learning and then in the second stage residual image is subtracted from input image which completes de-noising and Demosaicking. Experimental results of the MARN-JDD demonstrate that the proposed model incredibly surpasses many state-of-the-art joint denoising and demosaicking approaches on the basis of both peak signal-to-noise ratio (PSNR) and structure similarity index metrics (SSIM).

Key Words: *Adaptive residual network, denoising, demosaicking, data driven methods, convolutional neural networks*

Table of Contents

| | |
|---|------------|
| Acknowledgements | vii |
| Abstract | ix |
| Table of Contents | x |
| List of Figures | xii |
| List of Table | xiv |
| Chapter 1 | 15 |
| 1.1 Problem Statement..... | 15 |
| 1.2 Motivation..... | 15 |
| 1.3 Research Objectives..... | 15 |
| 1.4 Research Contribution | 16 |
| 1.5 Thesis Organization | 16 |
| Chapter 2 | 17 |
| 2 Denoising | 17 |
| 2.1.1 Spatial Filtering | 17 |
| 2.1.2 Linear Filters | 18 |
| 2.1.3 Mean filter | 18 |
| 2.1.4 Wiener Filter..... | 18 |
| 2.1.5 Non Linear Filters | 18 |
| 2.1.6 Median Filter | 19 |
| 2.1.7 Transform Domain Filtering..... | 19 |
| 2.1.8 Non- Data Adaptive Transform..... | 19 |
| 3 Demosaicking..... | 20 |
| 3.1.1 Color filter array patterns | 21 |
| 3.1.2 Color models | 22 |
| 3.1.3 Demosaicking methods | 23 |
| 3.1.4 Bilinear interpolation..... | 23 |
| 3.1.5 Iterative demosaicking..... | 24 |
| 3.1.6 Machine learning methods..... | 24 |
| 3.1.7 Adaptive color plane interpolation | 25 |
| 4 Neural networks | 25 |
| 4.1.1 Hidden layers..... | 25 |
| 4.1.2 Activation layers..... | 26 |

| | | |
|-----------------------|--|-----------|
| 4.1.3 | Training | 28 |
| 4.1.4 | Epoch..... | 28 |
| 4.1.5 | Regularization | 28 |
| 4.1.6 | Dropout layers | 29 |
| 4.1.7 | Loss functions..... | 29 |
| 4.1.8 | Norm based methods | 29 |
| Chapter 3..... | | 30 |
| Chapter 4..... | | 37 |
| A. | OVERVIEW | 37 |
| B. | MODIFIED ADAPTIVE RESIDUAL UNIT(MARU)..... | 37 |
| C. | NETWORK ARCHITECTURE DESIGN | 39 |
| D. | LOSS FUNCTION..... | 40 |
| E. | Training Strategies | 41 |
| 1) | Activation function | 41 |
| 2) | TRAINING Data..... | 41 |
| 3) | <i>Parameter Setting</i> | 41 |
| Chapter 5..... | | 43 |
| A. | Test Datasets | 43 |
| B. | Implementation Details | 43 |
| C. | Experimental Setting | 44 |
| D. | Comparison with state-of-the-art algorithms..... | 45 |
| E. | Running Time Comparison | 49 |
| Chapter 6..... | | 51 |
| Chapter 7..... | | 52 |

List of Figures

| | |
|--|----|
| Figure 2.1 artifacts in the tight diagonal pattern of a shirt's fabric. | 21 |
| Figure 2.2 Zipper artifact due to RF-feedthrough | 22 |
| Figure 2.3 The Bayer CFA. | 23 |
| Figure 4.1 Modified Adaptive residual unit (MARU)..... | 38 |
| Figure 4.2 The architecture of the proposed MARCNN-DDJ network..... | 38 |
| Figure 4.3 illustrates validation pnsr changes with 70 training epochs of the same network architecture. | 42 |
| Figure 5.1 Kodak per image results (PSNR) on noise level $\sigma = 20$ | 46 |
| Figure 5.2 Kodak per image results (SSIM) on noise level $\sigma = 20$ | 46 |
| Figure 5.3 McMaster per image results (PSNR) on noise level $\sigma = 20$ | 47 |
| Figure 5.4 McMaster per image results (SSIM) on noise level $\sigma = 20$ | 47 |
| Figure 5.5 Visual Results of McMaster18 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result (PSNR=24.80, SSIM=0.6084); (d) SEM result (PSNR=23.11, SSIM=0.4183); (e) DeepJoint result (PSNR=28.07, SSIM=0.7706); (f) ADMM result (PSNR=28.50, SSIM=0.7862); (g) GAN result (PSNR=30.00, SSIM=0.8387); (i) our MARN-JDD result (PSNR=30.164, SSIM=0.842)..... | 50 |
| Figure 5.6 Visual Results of McMaster17 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b)Original image; (c) FlexISP result (PSNR=22.80, SSIM=0.5217); (d) SEM result (PSNR=24.33, SSIM=0.4072); (e) DeepJoint result (PSNR=26.35, SSIM=0.7051); (f) ADMM result (PSNR=27.42, SSIM=0.7586); (g) GAN result (PSNR=28.33, SSIM=0.8000) (h) our MARN-JDD result (PSNR=27.66, SSIM=0.805)..... | 50 |
| Figure 5.7 Visual Results of McMaster7 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result (PSNR=25.62, SSIM=0.5647); (d) SEM result (PSNR=24.02, SSIM=0.3696); (e) DeepJoint result (PSNR=28.44, SSIM=0.7478); (f) ADMM result (PSNR=28.62, SSIM=0.7434); (g) GAN result (PSNR=29.63, SSIM=0.8134); (h) our MARN-JDD result (PSNR=30.31, SSIM=0.837)..... | 51 |
| Figure 7.1 Visual Results of McMaster4 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image;(b) Original image; (c) FlexISP result(PSNR=24.30, SSIM=0.6481); (d) SEM result(PSNR=22.67, SSIM=0.3639);(e)DeepJoint result(PSNR=29.04, SSIM=0.8819); (f) ADMM result(PSNR=28.89, SSIM=0.9119); (g) GAN result(PSNR=31.17, SSIM=0.9261); (h) our MARN-JDD result(PSNR=30.50, SSIM=0.92)..... | 58 |
| Figure 7.2 Visual Results of kodak24 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image;(b) Original image; (c) FlexISP result (PSNR=24.14, SSIM=0.5745); (d) SEM result (PSNR=22.79, SSIM=0.4939) ;(e) DeepJoint result (PSNR=27.30, SSIM=0.7925); (f) ADMM result (PSNR=27.43, SSIM=0.8082); (g) our GAN result (PSNR=28.82, SSIM=0.8460); (h) our MARN-JDD result (PSNR=29.32, SSIM=0.86)..... | 59 |
| Figure 7.3 Visual Results of kodak3 with $\sigma = 10$ noise for Joint denoising and demosaicking. (a) BayerNoisy image;(b) Original image; (c) FlexISP result (PSNR=30.90, SSIM=0.7521); (d) SEM result (PSNR=30.36, SSIM=0.6973) ;(e) DeepJoint result (PSNR=33.99, SSIM=0.9009); (f) ADMM result (PSNR=33.40, SSIM=0.8949); (g) GAN result(PSNR=36.57, SSIM=0.9370); (h) our MARN-JDD result(PSNR=35.28, SSIM=0.927)..... | 59 |
| Figure 7.4 Visual Results of kodak4 with $\sigma = 10$ noise for Joint denoising and demosaicking. (a) BayerNoisy image;(b) Original image; (c) FlexISP result (PSNR=29.67, SSIM=0.7395); (d) SEM result (PSNR=29.63, SSIM=0.7055) ;(e) DeepJoint result (PSNR=32.43, SSIM=0.8495); (f) ADMM result (PSNR=31.93, SSIM=0.8414); (g) GAN result (PSNR=34.27, SSIM=0.8928). (h) our MARN-JDD result (PSNR=34.45, SSIM=0.898)..... | 60 |
| Figure 7.5 Visual Results of kodak9 with $\sigma = 10$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result(PSNR=30.53, SSIM=0.7621); (d) SEM | |

*result(PSNR=30.71, SSIM=0.7244); (e) DeepJoint result(PSNR=34.01, SSIM=0.9031); (f) ADMM
result(PSNR=32.99, SSIM=0.9025); (g) GAN result(PSNR=36.05, SSIM=0.9280); (h) our MARN-JDD
result(PSNR=36.04, SSIM=0.925).....61*

List of Table

| | |
|--|----|
| Table 1 Activation functions for deep learning | 28 |
| Table 2 Deep learning papers review | 34 |
| Table 3 model architecture as a layer | 39 |
| Table 4 Average PSNR(dB) comparisons for the joint demosaicking and de-noise results. The best is in bold. | 48 |
| Table 5 Average SSIM comparisons for the joint demosaicking and de-noise results. The best is in bold. | 48 |
| Table 6 Running Time of Joint Demosaicking and Denoising (seconds) | 49 |

Chapter 1

INTRODUCTION

1.1 Problem Statement

Denoising subsequently demosaicking is untraceable, due to demosaicking distorts the attribute of the noise in a complex and hardly computable pattern. To guarantee the visual quality of reconstructed images particularly in the presence of noise corruption with efficient computation perceptivity to color artifacts in spaces of no or weak spectral correlation.

1.2 Motivation

Network depth is of essential usefulness in neural network architectures, but deeper networks are more difficult to train because of the notorious vanishing gradient problem as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitively small until now deep residual networks have not been applying demosaicking and denoising. On applying deep learning approaches on denoising or demosaicking process give positive results experiments performed. As in Deep learning techniques deep residual network outperformed other deep learning techniques. Proposed technique uses deep residual network on joint denoising and demosaicking process.

Despite being designed for the Bayer mosaic pattern, the methods proposed may be modified to apply to other patterns of mosaic.

1.3 Research Objectives

Applying Demosaicking on images having weak spectral correlation always needs efforts. Huge number of conventional Demosaicking techniques needs repetitive work along with multi-level interpolations with huge processing cost. Due to these things these techniques are not providing mature solutions for real time processing. This thesis proposed framework which is apply denoising conjunct with Demosaicking with

refinement using deep adaptive residual learning. Train the model to maximize the regularity found in natural images on a wide range of ground truth data.

1.4 Research Contribution

Our contributions to conjunct Demosaicking-denoising will be deep Adaptive Residual network competent of manipulation a wide scope of a procedure and noise levels to create a training set full of challenging images.

1.5 Thesis Organization

The thesis comprises six chapters each describing the techniques used in this research work. In the first chapter, a problem statement, introduction, motivation and, objective and scope of study were given. The second chapter comprises of background of denoising, demosaicking and neural network. The third chapter contains literature survey to understand the merits and demerits of existing joint demosaicking denoising methods. The fourth chapter deals with the proposed algorithm called Modified adaptive residual network joint demosaicking and denoising (MARN-JDD) that de-mosaick and de-noise from an image. In chapter five, the proposed algorithm result is comparable in quality and time to the state-of-the-art algorithm. The concluding chapter describes the strengths and weaknesses of this work and future extension work.

Chapter 2

BACKGROUND

This portion provide brief background of demosaicking, denoising issues along with neural networks and their unit.

2 Denoising

De-noising is a digital image process utilized to removing unwanted noise to be able to bring the original image. There are several noise reduction algorithms in image processing. When picking an algorithm for noise removing, several factors need be weighed:

- the usable time and computer power:
- a digital camera must use a tiny onboard CPU to reduce noise in a fraction of a second, while a computer's desktop has more time and power.
- whether that is justifiable to sacrifice real detail if more noise can be eliminated (how competitively it is possible to determine where to go).

There are three essential techniques to de-noise image – domain filtering, spatial filtering and wavelet thresholding. Every filtering method's objective are:

- Effectively abolish uniform noise areas.
- Edges and similar image preservation.
- Maintain natural visual presence.

2.1.1 Spatial Filtering

Using spatial filters is a conventional approach of removing image data noise. Spatial filtering is the preferred method in cases where there is only noise of additives. It can also be grouped into two classifications: linear filters and non- linear filters

2.1.2 Linear Filters

In scenarios where there is only additive noise, that's the preferred method. The ideal linear filter for Gaussian noise is a square error mean filter. It fits acute boundaries, destroys image lines and details of fine images. Wiener filter and Mean filter are involved. The ideal linear filter for Gaussian noise is a medium square error filter. Linear filters tend to demolish lines, blur sharp edges as well as other fine image details and fail in signal-dependent noise. The Wiener filtering process involves information on the original signal and noise spectrum and when the underlying signal then only works smoothly. The Wiener method pursues spatial smoothing and the window complication is controlled by its models. Johnstone and Donoho proposed a denoising strategy based on wavelets to resolve the Wiener filtering vulnerability.

2.1.3 Mean filter

It is a filter that gives a smooth image by decreasing the adjacent pixel variations in intensity. The mean filter is actually an averaging filter. On each pixel the mask is enforced in the signal. Thus, each pixel component that falls under the mask is average filter for making a single pixel. The main drawback is that the edge preservation criteria in the mean filter are low.

2.1.4 Wiener Filter

It is a filter that approaches the filtering of noise statistically that has corrupted a signal. This filter can acquire the favored frequency response. The Wiener filter is approached from another angle to carry out the filtering process, the spectral attributes of the original signal and the noise must be known, to achieve the benchmark, the LTI filter with output as near to the real signal as possible.

2.1.5 Non-Linear Filters

Nonlinear filter is the preferred strategy in positions where there is a multiplying and functional noise. Noise is eliminated without a non-linear filter explicitly identifying it. Spatial filters use a low-pass filter in pixel groups, which implies a higher frequency

spectrum region. Spatial filters generally eliminate noise to an acceptable degree but at the expense of blurred images that make the edges in images invisible. To overcome this disadvantage in latterly. A number of nonlinear median filters have been developed, such as weighted median, relaxed median and conditioned rank selection, noise can be evacuated without exclusively identifying with non-linear filters. The median for the neighborhood pixels determines in this case, the output pixel value. Spatial filters on pixel groups use low- pass filtering to state the noise is in the region of the higher frequency spectrum.

2.1.6 Median Filter

Median filters are categorized as non - linear filters. First find the median value around the window and replace the median pixel value with each entry in the window is called Median filtering. If a strange number of entries are in the window, it is easy to define the median: After all entries are sorted numerically in the window, it is only the middle value. But there is more than one possible median for even a number of entries. It's a sturdy filter. Median filters used to provide smoothness in the processing of images and time series. Median filtering has the advantage that it is far less unstable than the average to extreme values (known as outliers). The outliers can therefore be removed without lowering the image's sharpness.

2.1.7 Transform Domain Filtering

It can be separated as per the core functions selected. They primarily categorized non-data adaptive transformation and data adaptive transformation

2.1.8 Non- Data Adaptive Transform

a. Spatial Frequency Filtering

Spatial Frequency Filtering assigns rapid low - pass filters to Fourier Transform. By selecting break frequency and adjusting a frequency domain filter and If the noise items are decorated with a convenient sign, noise is eliminated. The essential drawback of Fast Fourier Transformation (FFT) is that the edge information is not located in time or

space and is distributed over frequencies due to the FFT base function, that implies that time information is lost and edge streaming results from low-pass filtering. However, Wavelet Transform's localized nature in space and time is an especially effective way to de-noise images when it is important to preserve the edges in the scene.

b. Wavelet Domain Filtering

Wavelet domain is favored in working as the Discrete Wavelet Transformation (DWT) concentrates the signal energy in limited coefficients, so that the noisy DWT image comprises of a limited number of a high signal-to-noise ratio (SNR) coefficients, whereas a low SNR of a relatively large number of coefficients, the image is reconstructed using the reverse DWT after the low SNR coefficients (i.e. noisy coefficients) are removed. In consequence, from the observations, noise is filtered or eliminated. Wavelet methods have a major advantage in providing frequency localization and time at the same time. In addition, Wavelet methods define these signals far more efficiently than the original domain or transform them with global elements like the Fourier transformation.

3 Demosaicking

The demosaicking is a process in digital image processing that produces complete image from the incomplete or limited data which produces from the image sensor superimposed alongside an array of color filters. To obtain three full-channel color images directly from complementary sensors for metal-oxide-semiconductors (CMOS) or 3 charge-coupled device (CCD) or both can be used, this way, however, is expensive and still requires spectral band passes, sometimes in the kind of beam splitters alongside half-transparent mirrors. A CFA for capturing one color component per pixel in front of the sensor and interpolating missing color components is a more cost-effective solution, market also uses the most digital camera. To separate the color information CFAs are used, since light intensity only detected by the typical photo sensor without information about wavelength. In fixed patterns the CFA consists of spectrally selective filter arranged. CFA images or mosaic images are sparsely sampled color channels collected from the color filter array.

3.1.1 Color filter array patterns

The Bayer pattern [49] is the best widely used CFA [50]. Image artifacts caused by demosaicking CFA's. Aliasing is a common reason for image artifacts, it happens when the signal is sampled at the highest frequency in the signal at less than twice as high. Low sampling of images causes aliasing in the spatial domain.

It occurs when a repetitive pattern of high spatial frequency is sampled at low frequency, which is a large-scale interference pattern known as Moiré patterns. An unrealistic maze-like pattern color artifact or pixels arranged; the interference appears as repetitive patterns. In figure 2.1, Example of Moiré artifacts can be seen. The false color artifacts can also be caused by demosaicking. Across the edge the interpolation might have been done, these often manifest themselves along edges. The commonly occurring along edges is zipper artifacts. The demosaicking algorithm averages pixel values over an edge when zippering is prevalent, in a zipper pattern become blurred the edges. In figure 2.2, a zippering example can be seen.



Figure 0.1 artifacts in the tight diagonal pattern of a shirt's fabric.



Figure 0.2 Zipper artifact due to RF-feedthrough

A CFA pattern measuring the green image on a grid with half the image resolution and red and blue images on rectangular grids with a quarter of the image resolution is known as a Bayer pattern, see figure 2.3. At a higher sampling rate, the green channel is measured. The human visual system is more sensitive to the green wavelength. In most color filter arrays above the Bayer pattern this property is used. The previous demosaicking work covers only techniques for the Bayer pattern, it is common to use. The RGBW Bayer pattern of Sony and the Clarity+ pattern of Aptina are two more widespread color filter arrays. The same structure as the Bayer pattern is replaced with clear filters that make the pixel's register light of all wavelengths except for any other green filter is known as Sony RGBW pattern. In rectangular grids this provides a pattern with all four different kinds of filters. Except for all green filters, the same structure as the Bayer pattern is replaced by clear filters is known as Aptina Clarity+ pattern.

3.1.2 Color models

Frequently, by the way of different color coordinate systems colors are described and organized. The most used and popular color model is the RGB color model in which the color is expressed as an additive blend of the three primary colors red, green and blue. The YCbCr model is another widely used representation. They are decorrelated and detached into a channel of luminance and two channels of chrominance. Linear blend of

red, green and blue channels is build up by the luminance channel and the weighted variation between the luminance channel and the red jointly blue channel as referred as the chrominance channels.

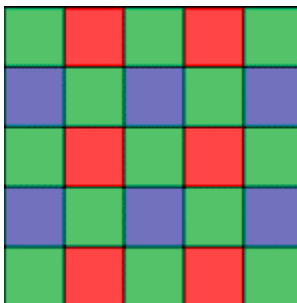


Figure 0.3The Bayer CFA.

3.1.3 Demosaicking methods

It is generally completed over interpolating the missing color channel information is Demosaicking. The following portion define important demosaicking in spatial dimensional performing techniques.

3.1.4 Bilinear interpolation

The least complex bilinear (or straight) technique for interpolation essentially assess the pixel esteem as the mean of each shade of the nearest pixels, for example Four qualities are utilized to assess the green pixel estimation of a non-green pixel, while a red or blue pixel esteem just approaches two nearby pixels in the right shading. This straightforward method does not utilize the relationship between various shading layers and consequently creates obscured pictures inadequately. The procedure can be reached out to incorporate more pixels, including the pixel value, in the present pixel. A case of this is a Malvar et al method. [51] in which 9 pixels are utilized to assess the green channel and 11 to gauge the red and blue channels. This improves the connection between's the shading channels. It improves the connection of shading channels. So as to calculate the pixel value, a weighted aggregate of the comparing pixels is determined by weight.

A mosaic image is green with twice the number of pixels (and also more details) as the blue or red portion. One prevalent way is interjecting the green (or luminance) channel first and after that measure the red and blue (or chrominance) channels utilizing the

totally added green channel[50]. The reason is to prevent aliases which occur less likely in the green channel because they encode as many information twice. Li et al., as defined. [50] An underlying edge recognition calculation is frequently used to insert edges to wipe out zippering effects along the edges. The chrominance channel is added by going up against a consistent tone and evaluating the contrast between the blue and green channels and the red and green channels. The pixel green channel esteems are deducted and the Red and Blue channels are assessed. The downside of this procedure is commonly that the Green channel interpolation engenders mistakes into the red and blue channels that can at last make major demosaicking picture blunders. Another procedure is to insert the luminance channel where the missing information is added by heuristic edge-direct principles in the green channel The course of the neighborhood edge is evaluated from the accessible information on the green, red or blue channel and the inclinations of the second request of the chromium channels can be utilized as a rectification. Nearby covariance can be utilized to modify interpolation, assessed based on geometric duality.

3.1.5 Iterative demosaicking

It has a basic flaw in the propagation of errors where any errors that occur inevitably during the interpolation of the luminance channel spread to the chromium channels Sequential demonstration. This can be solved by iterative redevelopment. According to the rules of color ratio are refined green, red and blue channel.

3.1.6 Machine learning methods

In the field of image processing When machine learning methods obtain additional ground, a number of various machine learning techniques have been attempted. Neural networks with convolutionary and multi-layered kernels proved helpful and may be used both for demosaicking and denoising when properly designed Super resolution [2] is an nonlinear method of increasing the resolution by interpolating new values between existing pixels nonlinearly. In order to find the missing values, it is possible to channel this approach to a mosaic image. Generative opposing networks [33] are a successful method of machine learning for tasks with super resolution. The generative adversarial

network is a network structure with two parallel networks, one generative network and one discriminative network, which are together prepared to improve the errands required.

3.1.7 Adaptive color plane interpolation

A demosaicking method using the adaptive color interpolation plane was introduced in 1996 by Hamilton and Adams [49]. A sequential approach to demosaicking split the image into RGB color channels is used. The green channel was first interpolated, and the red and blue channels interpolated. Interpolation is carried out by an edge detection approach that allows interpolations to be carried out on and not across borders.

4 Neural networks

It is influenced by the way neurons function in the brain, which only signal when specific conditions, i.e. are met. In order to produce an output, a combined neuron input must exceed certain threshold value (bias). In 1943, McCulloch and Pitts [52] proposed a "threshold logic"

$$y = f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^N w_i x_i \geq b \\ 0, & \text{if } \sum_{i=1}^N w_i x_i < b \end{cases} \quad (\text{Error! Bookmark not defined.})$$

Where w_i is a weight, x_i is the value for the input, and b is the values for the threshold. $Y=1$ is activated for the neuron. Neural networks are ideal for several computer functions like classification and clustering of neural networks and the use of neural networks for various types of mathematical and computer operations has increased because they have useful features.

4.1.1 Hidden layers

A neural net is comprised of neuron layers that connect a layer input to the preceding layer output. Input layer, one or more layers hidden comprise the simplest neural grid and output layer. The covered layers contain learning weights and biases optimized by back propagation, which is a two-phase cycle propagation to update neural network weights. The weights determine the amount of each neuron input and the bias defines the

value of the threshold. Hidden layers can be completely connected, and all input variables can be connected to all output. They may also include convolutionary layers or local layers

4.1.2 Activation layers

Hidden layers frequently have a non-linear layer, for example, a sigmoid or linear rectifier (ReLU) work. The system enactment work makes the system nonlinear and progressively mind boggling. On the off chance that the initiation capacities were excluded, it is conceivable to diminish the system to linear regression.

In the equation 6 it is shown that the easiest step function activation feature. The function ReLU is defined as

$$f(x) = \max(0; x) \quad (2)$$

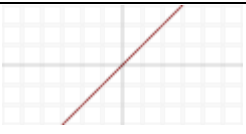

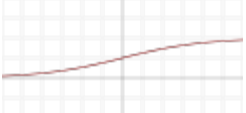
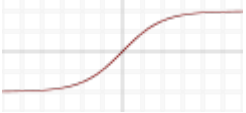
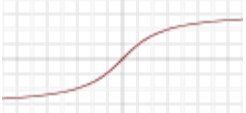
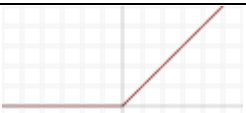

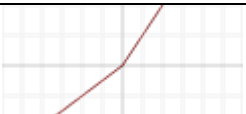
is presently the most frequently used neural network activation function [43]. A number of recent studies, including Krizhevsky et al. [53] exhibited that the ReLU work expands the neural system union contrasted and the sigmoid capacity. The broken ReLU and parametric ReLU capacities are created. The flawed ReLU adds a little incline to the negative part and characterizes it as

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x \leq 0 \end{cases} \quad (3)$$

Where a is a constant which is set to small value. The parameter ReLU is characterized similarly as a parameter can be prepared. The sigmoid capacity is characterized as a condition. The sigmoid capacity has an esteem district of $[0;1]$ that gives it a mean esteem that isn't zero. The tanh work is like and characterized as the sigmoid capacity.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

The tanh function has an $[-1;1]$ esteem area and a normal estimation of zero. The functional forms sigmoid, tanh and ReLU are shown in table 4.

| Name | Plot | Equation |
|--|---|--|
| Identity |  | $f(x) = x$ |
| Binary Step |  | $f(x) = \begin{cases} 0, & \text{for } x < 0 \\ 1, & \text{for } x \geq 0 \end{cases}$ |
| Logistic |  | $f(x) = \alpha(x) = \frac{1}{1 + e^{-x}}$ |
| TanH |  | $f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$ |
| ArcTan |  | $f(x) = \tan^{-1}(x)$ |
| Rectified Linear Unit(ReLU) |  | $f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$ |
| Leaky rectified linear unit (Leaky ReLU) |  | $f(x) = \begin{cases} 0.01x, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$ |
| Parametric Rectified Linear |  | $f(a, x) = \begin{cases} ax, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$ |

| | | |
|-------------|--|--|
| Unit(PReLU) | | |
|-------------|--|--|

Table 1 Activation functions for deep learning

4.1.3 Training

In order to determine the net's ultimate weight and bias, it must be trained to produce as much output as feasible as a recognized image of ground truth, for example. The image to be reproduced by the network through demosaicking. This is achieved by an algorithm of optimization. The optimization algorithm in other words reduces the loss function. Quantify the difference amongst the ground truth and net result. The loss function gradient is calculated to determine the direction in which the optimization algorithm should update weights. Ideally, iterates the process till the outcome converges. If the image produced is the same as the image of the ground truth and the training is iterated until the loss function produces a sufficiently low value, true convergence is difficult to achieve.

4.1.4 Epoch

An epoch is a way of measuring of where the entire training data is once passed through the neural network. A neural network requires training for several periods in order to achieve convergence with limited data. As Kapah et al. has done, fixed number can be set as number of time and also variable number can be used for this over iteration. [54], as training continues to meet certain termination criteria, as Gharbi et al. [2] have done.

4.1.5 Regularization

Neural networks can be over-fitted if the network does not perform well on real data but performs pretty good when it is train on training data. To avoid this effect, regularization of weights is general. It implies that larger weights are penalized and the process of training to find lower weight values is encouraged. A regularization technique can have defined as

$$\check{E}(w) = E(w) + \lambda \cdot \frac{1}{2} w^2 (5)$$

In the equation E shows the loss function, w shows the weight and λ refers to the constant that is very small. While using in neural network theregularization is then used as weight decay.

4.1.6 Dropout layers

Another way to avoid the effect of overfitting is to apply a drop-off the layer to every layer. With a certain number of the neurons and zeros the drop-out layer interferes with this number. By this process the neurons are discharged and distributed randomly and changed for each and every training data batch. It can likewise be viewed as different systems with different neural associations that are sub-sampled and prepared in the meantime. All neurons are utilized to deliver a yield that accepts the diminished qualities as zero qualities into record. Dropout layers are not utilized during validation.

4.1.7 Loss functions

To quantify the convergence and the difference between the net output and the ground truth is known as loss function. Over the years, Numerous techniques have been proposed, both straightforward systems utilizing standard qualities and further developed methods utilizing results-based connection with ground truth.

4.1.8 Norm based methods

The L2 Loss function is a common loss function for evaluating neural networks. The method simply calculates the Euclidean norm square and can distinguish the result from the ground truth and can be described as

$$L_2(x, y) = \sum_{i=0}^n (y_i - x_i)^2 (6)$$

Chapter 3

RELATED WORK

Demosaicking and de-noising are the most important fundamental and critical stages in digital image pipelines. Demosaicking refer as procedure to rebuild an entire color image from partial color information created with an overlaid sensor image having a color filtration array in digital image method. De-noising is a digital image process utilized to removing unwanted noise to be able to bring the original image. According to assessment, the present information of color image (CI) is adulterated due to two-third and noise of CI data is inadequate. To achieve modularity, denoising and demosaicking techniques are managed in a successive way or independently. This direct leads to error accumulation because Demosaicking used the imitation of unreliable samples and denoising results get perturbed due to variable and non-linear per pixel interference launched by Demosaicking due to utilization of unreliable samples. It's been apperceived with time which exploiting the regularity of natural images is crucial to hoisting beneath guarantee. Interpolation predicated techniques such as for instance spline interpolation, bi-cubic interpolation, and bilinear interpolation, had been the first methods suggested to handle the demosaicking quandary.

These techniques are able to calculate the missing color values however they also engender unwanted color artifacts including zippering, purple fringing, chromatic aliases, then blurring. Numerous approaches have been proposed [12] - [24] to cope with mendacious color artifacts and get high resolution images. Residual interpolation strategies suggested by Kiku et al. [19] [20] [21] [22] provide better quality images than straightforward interpolation-based methods exhibiting the possibility of recurring estimation for image demosaicking.

Machine Learning based demosaicking recently started to attract increasingly more interest in recent years. Early works (e.g., [25] and [26]) with an easy fully connect network just achieved success that is limited; eventually works depending on Support Vector Regression [27] or Markov Random Fields [28] were capable of attaining similar functionality to model based demosaicking. Most recently, the area of rich learning or even deep neural networks has progressed quickly triggering breakthroughs in both low-level and high-level vision problems [29] - e.g., image recognition [30], [31], deal with recognition [32], image super resolution [33] as well as impression denoising [34]. By comparison, image demosaicking by serious learning has stayed a primarily unexplored territory with the exceptions of [35] and [36]. So, it's normal to leverage latest developments in deep learning on the area of image demosaicking for more improvement. Zhang and others [37] and w. Yang et al. [32] used network-based deep learning techniques to restore images in high-quality images. This motivated us using an architecture that relies on convolutionary network layers with residual image denoising and demosaicking estimates.

Numerous strategies for Demosaicking and denoising techniques have been proposed within the literature.

A joint strategy to Demosaicking and denoising proposed by Buades et al. 2009]. Heide et al. [2014] by enclosing a non-local all-natural image preceding into a boost approach. Be that as it can that their previous is still handmade and the mixture of boost. The dependency of nonlinear pounds for each one interpolant is on the inter channel color alterations for the corresponding direction, as a result of the divergence of inter channel color variations for every direction is indicative of the probability based on the place of an edge [10].

Gharbi et al. proposed a brand-newmethod to resolve the jointDenoising andDemosaicking using the convolutional neural network (CNN) images while reducing distortion in images like Moire and zippering. Proposed scheme has performed very well in flat images nevertheless its overall performance is decreased in tough cases on account of 2 issues that are important. First, tough events are unusual for weakened by the immensely more popular simple areas. Second, metrics like as PSNR or L2 don't recognize Demosaicking artifacts which are salient to people. 3 steps learning approach

are used by it. Firstly, criterion database of pictures is used by them and also teach the system with this data. Secondly mining the cold hard patches and create a dataset of challenging pictures. Finally, Retrain the device on the cold hard patches by retraining the device on tough patches, the functionality on the method increases. A deep neural network is train to demosaick challenging images with luminance artifact around small buildings, e.g. color Moire artifacts and zippering effect. The methodology appropriately gets a handle on the convoluted examples and yields contortion less outcomes. The breaking points of the methodology are that it controlled by distinguishing testing spots for use in preparing obviously, if ground truth pictures as of now contain ancient rarities like Moire, the system would have figured out how to incorporate curios in image [2].

Demosaicking uses a multilayer neural network Wang [8] proposed a 4 by 4 patch dependent multilayer neural network was used for image Demosaicking. In contrast with cutting edge Demosaicking algorithms the multilayer neural system oversees unexpected shading changes legitimately, be that as it may it springs up inadequate amid recovering colossal repeat designs. A theory was that using bigger patches would improve the system recuperate path for high repeat designs.

Demosaicking making utilization of artificial neural systems Kapah et al. [9] early analyzed Demosaicking with the assistance of counterfeit neural systems. The perceptron, the backpropagation form, the selector displays together with the quadratic perceptron demonstrate had been set nearby each other. It was found that the perceptron was truly adept at Demosaicking exceptionally low recurrence territories and fizzled at high recurrence regions which have immersed hues. Contrariwise the backpropagation display was great at Demosaicking high recurrence regions and furthermore at improving shading think about despite the fact that in lower recurrence areas it neglected to recreate the best possible hues. The selector show was fit the bill to be able to pick once the perceptron and backpropagation demonstrate had been going being used to demosaick the region of an image. Thusly, high recurrence regions could be demosaicked making utilization of the backpropagation style and diminished recurrence areas were demosaicked making utilization of the perceptron. The last methodology which was inspected was the quadratic perceptron plan which preformed fantastic in each low and high recurrence district and furthermore played out the best among the analyzed techniques.

The suggested methods use 4 directional high-order interpolants of the missing color value determined in 4 different directions to maintain the edge in Demosaicking [5]. Each interpolant is then given non-linear weight. Each nonlinear mass is derived on the basis of the color differences between the channels, in order to increase the impact of the interpolant from a similar angle aspect, while reducing the effect of the interpolants on the exact opposite aspect of an advantage. This allows the suggested algorithm to improve the accuracy of interpolation in soft areas while minimizing color artifacts on the edges.

To obtain a full color image a method is proposed [1] that linearly combines an extracted luminance image and RGB image who's passed by low pass filter. In order to denoise the luminance picture and minimal passed RGB image, this particular pattern modifies the non-local mean before combining these images. It assumes the R, B and G part of image have a similar look in a pure image.

In order to prevail over issue within the multi spectral that the proposed method uses adaptive multispectral Demosaicking dependent on frequency domain analysis [6]. Color filter array interpolation or maybe 3 band Demosaicking could be the procedure to locate the missing color sample from RGB bands to reconstruct the whole image. though it resolved a difficult problem of multispectral Demosaicking, wherein every band is considerably under sampled as a result of the increment in the quantity of bands. It proposed an adaptive spectral correlation

Table 2 Deep learning papers review

| Paper | Used for | Dataset | Network depth | Kernel size | Activation function | Filters | Optimization | Initial weight strategies |
|--|---|--------------------------------------|---------------|-------------|------------------------------------|---------|---------------------------------------|---------------------------|
| Deep Joint Demosaicking and Denoising [2] | Joint denoising and demosaicking | Imagenet, MirFlickr | 17 | 3*3 | ReLU | 64 | SGD and Adam | SGD |
| COLOR IMAGE DEMOSAICKING VIA DEEP RESIDUAL LEARNING [36] | Demosaicking | Kodak and McMaster | 10 | 21*21 | ReLU | 64 | gradient descent algorithm | SGD |
| Image Demosaicking and Blind Denoising [47] | Joint Demosaicking and denoising | Kodak and McMaster | 16 | 3*3 | Relu Maas et al. (2013) activation | 64 | n/a | SGD |
| Iterative Residual Network [46] | Joint Demosaicking and Denoising | McMaster, Kodak and MIT dataset | 5 | 5 × 5 | PReLU | 64 | stochastic gradient descent algorithm | SGD |
| Joint Demosaicking and Super-Resolution [45] | Joint Demosaicking and Super Resolution | RAISE | 26 | n/a | PReLU | 256 | ADAM | SGD |
| Adaptive Residual Networks [37] | Image Restoration | Berkeley Segmentation Dataset BSD500 | 8 blocks | n/a | PReLU | 64 | ADAM | Xavier Initialization |

| | | | | | | | | |
|------------------------------------|--------------|--|-------|--------------|------------------|----|-----------------------------------|-----|
| DeepDemosaicking: [44] | Demosaicking | BSDS 500 dataset, ARRI dataset, SuperTex136 dataset, General-100 | 7 | 5×5 | Not specified | 64 | n/a | SGD |
| Beyond a Gaussian Denoiser [34] | denoising | Berkeley segmentation dataset, CBM3D | 17,20 | 3×3 | ReLU | 64 | SGD and Adam for comparison | SGD |

based Demosaicking that utilizes a novel along with anti-aliasing filter to lessen the interference. The proposed scheme incorporated an intra prediction program to make a far more precise reconstruction of the image as we encounter in Demosaicking and also denoising **case**, when a tough image will come and proposed algorithm doesn't work subsequently the algorithm learns from tough image and in future when the tough image is of that kind and then the proposed algorithm works very well.

The purposed strategies try and boost reconstructed image quality by taking into consideration inter channel correlation via utilization of Demosaicking strategy. It's a multispectral Demosaicking algorithm in that the method uses a lot more spectrum of color of the image than the RGB picture. This method makes an attempt to lengthen Demosaicking process via color impact algorithm to think about inter channel correlation. The foundation of the Demosaicking algorithm is on Brauers Color Difference (BCD). By utilizing BCD throughout the mixture of goals as well as guide bands are interpolated uniformly [3].

As stated in [2], denoising and demosaicking will often be addressed as 2 separated issues as well as analyzed by diverse communities. Within training, raw CFA information are usually polluted by sensor interference [11], that could result in unwanted artifacts to come down with reconstructed images in case unattended. Ad-hoc sequential methods concatenating two functions frequently fail: 1) denoising prior to demosaicking is tough because of unfamiliar racket qualities as well as aliasing created by the CFA; denoising right after demosaicking is difficult also simply because interpolating CFA would complicate the racket conduct within the spatial url (e.g., getting signal dependent). Thus, conjunct demosaicking as well as denoising (CDD) happens to be conceived a far more adequate method of trouble formula. Since each denoising and demosaicking are ill posed, a typical airier dependent image prior could be brought to facilitate the answer to CDD. Before as well as subsequent to demosaicking denoising applied.

Chapter 4

PROPOSED MODIFIED ADAPTIVE RESIDUAL Model (MARCNN - DDJ)

A. OVERVIEW

In this section, author proposed a 2-step standard feed-forward network architecture MARCNN - DDJ to perform demosaicking and denoising with the help of proposed adaptive deep residual CNN model. In general, the training of a deep CNN model for a particular task comprises two steps: (i) design of network architecture and (ii) model learning from training data. For the design of network architecture, we adjust the VGG network [42] with customized adaptive residual block [37] to make it feasible for denoising and demosaicking of images and to determine the depth of the network used in state- of- the- art demosaicking and denoising techniques. Authoradopt the residual learning formula for model learning and incorporate it for fast training and improved demosaicking and denoising performance with batch normalization. In the following sections, we will elaborate implementation details including network architecture design, loss function, and training procedure.

B. MODIFIED ADAPTIVE RESIDUAL UNIT(MARU)

Modify the adaptive residual block [37], removing scaling factors, change original input vector to previous layer output vector and add batch normalization between convolutional and activation function i.e.PReLU[43].

The technique uses modified adaptive residual unit, as shown in Fig. 4, can be formulized as:

$$y = F(x, Wi) + x \quad (7)$$

Here $F = W_2\sigma(W_1x)$ in which σ denotes PReLU, x_1 indicates the previous layer output vector, proposed technique use MARU as can be seen in Fig.4.

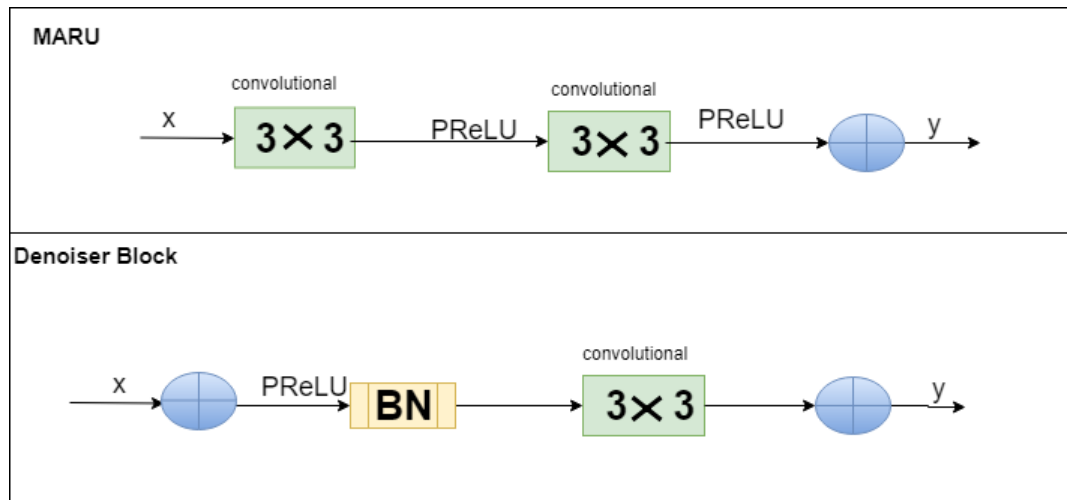


Figure 0.1 Modified Adaptive residual unit (MARU).

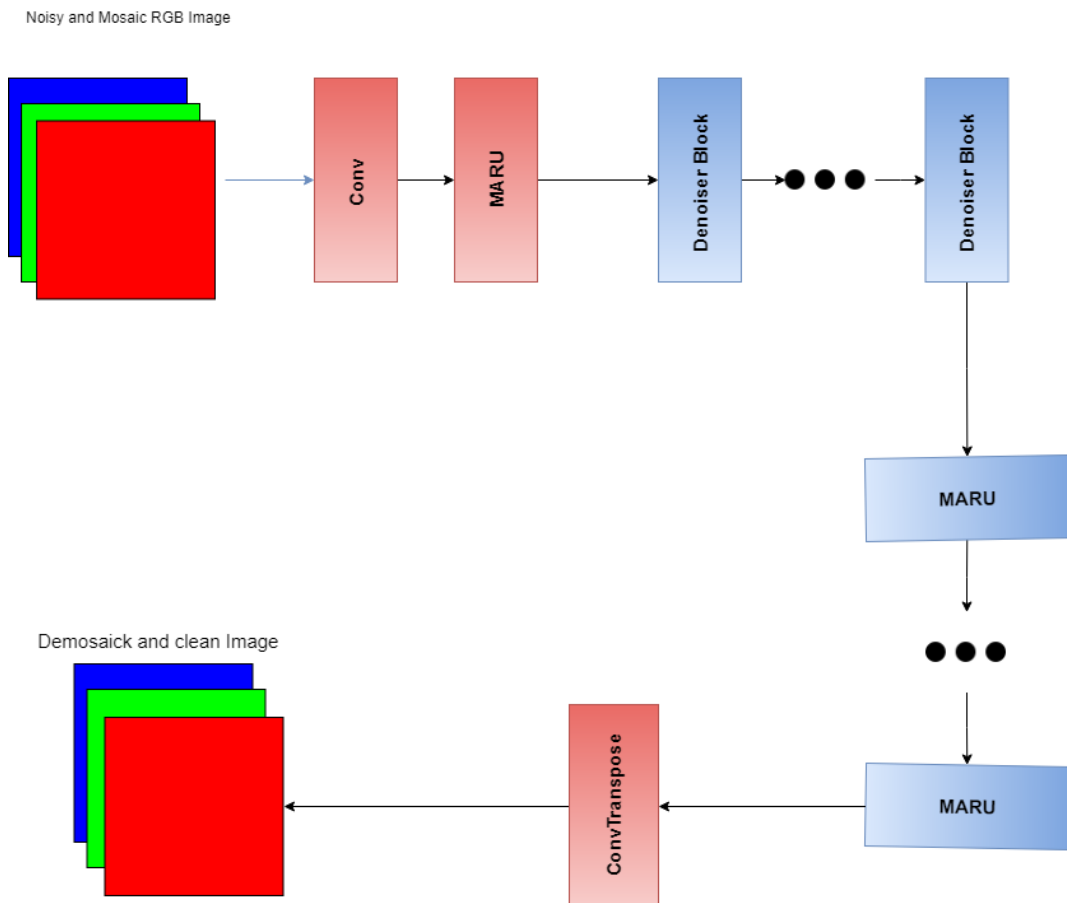


Figure 0.2 The architecture of the proposed MARCNN-DDJ network.

MARU block has three differences compared to the adaptive residual unit [3]. First, we add normalization of the batch between convolution and activation, i.e. PReLU. This is important to denoising and improve performance. Secondly, our shortcut connects the outputs of the previous layers to the current layer instead of the original input with the following stacked layers. As a problem, the useful input information

does not need to be reserved. Third, we remove α and β because this problem does not need to be scaled.

C. NETWORK ARCHITECTURE DESIGN

The input of proposed ADRCNN is a noisy observation $y = x + v$. As can be seen in Figure 4.2, our MARN consists of one 3×3 convolutionary layer, one MARU, $d-1/2$ De-noiser units, $d-1/2$ MARUs and one 3×3 convolutionary layers sequentially. The first 3×3 convolutionary layer is used to extract the features from the noisy mosaic images input. The extracted features will then be forwarded to the MARU. The extracted features are subsequently sent to the $d-1/2$ De-noiser units. The extracted features are subsequently sent to the $d-1/2$ MARUs. and the last 3×3 convolutionary layer is then used to reconstruct noisy-free and demosaick image. Facts have shown that 3×3 layers of convolution [30] are sufficiently capable of extracting good features as long as the network is sufficiently deep.

Table 3 model architecture as a layer

| layer | filter(size) |
|-------|----------------|
| conv1 | 64(3×3) |
| conv2 | 64(3×3) |
| | 64(3×3) |
| | . |
| Conv3 | 64(3×3) |
| | 64(3×3) |
| | . |
| | 64(3×3) |
| Con4 | 64(3×3) |
| | residual layer |

In accordance with the basis defined in [42], all layers of pooling are removed and the size of convolutionary filters is set to 3×3 . Given the MARCNN - DDJ with depth 20, there are four kind of block, shown in Fig. 5.

(i) Conv+PReLU: To extract the features of the input mosaic noisy images, the first convolutionary layer of 3×3 is used. For the first layer, 64 size $3 \times 3 \times c$ filters are used to generate 64 feature maps, and for non-linearity, parametric rectified linear units (PReLU, $\max(0, \cdot)$) are used. Here c is the number of image channels, i.e. $c=1$ for gray images and $c=3$ for color images.

(ii) Conv+BN + PReLU (DenoiserBlock): for layers $2 \sim (D/2)$, 64 filters of size $3 \times 64 \times 64$ are used, and batch normalization is added between convolution and PReLU and subtract residual from original image.

(iii) Conv+PReLU+Conv+PReLU (MARU): for layers $D/2 \sim (D-1)$, 64 filters of size $3 \times 64 \times 64$ are used.

(iv) ConvTranspose: last 3×3 convolutional layer works as reconstructing noisy free complete images.

In many low-level vision applications, the yield image measurement should most often be the same as the original. This can lead to artifacts on the boundary. We pad zeros in particular to ensure that each feature map of the middle layers is indistinguishable from the original image. We find that no boundary artifacts result from the basic zero padding procedure. This great property is probably due to the MARCNN-DDJ's dominant capacity.

D. LOSS FUNCTION

We adopt MSE as the loss function to support the efficiency of the proposed model. To learn the end-to-end function F from incomplete and noisy images to its noisy free complete image counterparts, the weights represented by the shortcut kernels must be estimated. This is achieved by reducing the MSE between the network outputs and the original image. In particular, given the collection of n training image pairs x_i, y_i , where x_i is incomplete and noisy images and y_i is the noisy free full image version, we minimize the objective function:

$$L(\theta) = MSE = \frac{1}{n} \sum_{i=1}^n \|F(x_i, \theta) - y_i\|^2 \quad (8)$$

E. Training Strategies

1) Activation function

After conventional sigmoid-like units used in deep learning, ReLU was a popular activation function. For ReLU, the result is zero if the input is less than zero. PReLU introduces a learning parameter in the negative part. It's a general form of ReLU, of course. As stated in [43], the slope an in PReLU is an adaptively learned parameter that can offset the positive mean of ReLU and make it somewhat symmetrical. Experiments also show that PReLU converges and performs better faster than ReLU.

2) TRAINING Data

The training data plays an important role in deep learning. It is broadly known that deep learning generally benefits from the accessibility of huge scale training data. In order to prepare the proposed network successfully, we need to collect sufficient high-quality full color images as training data. Data sets the thesis using isMSR-demaicking [48]. The Microsoft Research Cambridge demo data set consists of a set of raw images and their downscale versions that can be used in linear space and color space to learn and evaluate demosaicking (and possibly other tasks such as denoising).

3) Parameter Setting

In our experiments, the network weights are initialized based on the method in Xavier Initialization. The Adam optimization is adopted to optimize the network parameters Θ

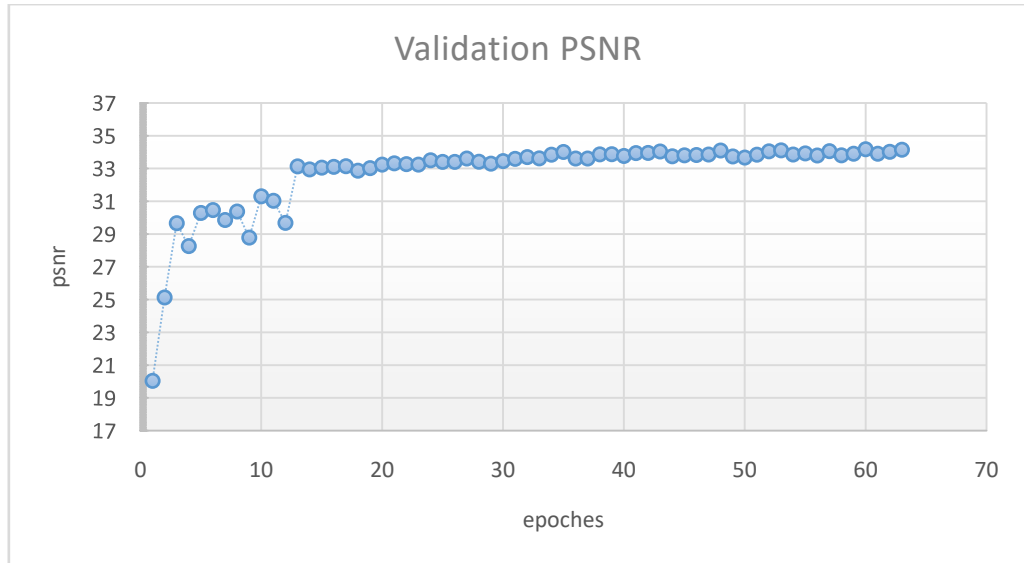


Figure 0.3 illustrates validation psnr changes with 70 training epochs of the same network architecture.

Figure 4.3 show validation set mean psnr for each epoch. Validation mean psnr start from 20 to 34.95. for first 15 epochs psnr increase with marginally gap and at some point, validation mean psnr decrease. After 15 epochs psnr increase with marginally minor gap but important increase. MARN first epochs mean validation set psnr is high as MARN is using xaivor initialization for initialization weight.

Chapter 5

EXPERIMENTAL RESULTS

In this section, we explain the details of the implementation and examine the quality of the proposed MARN - JDD method by comparing it to standard and state - of - the - art algorithms for joint demosaicking and denoising.

A. Test Datasets

We compare the proposed method with state-of-the-art Kodak data set and McMaster data set, which contain respectively 24 and 18 natural images with a resolution of 500x500. The images of Bayer are sampled from images of ground truth. Noisy images are obtained by adding Gaussian white noise from ground truth images. The noise is homogeneous if not specified. We chose Kodak due to each Kodak dataset used for comparison with denoising - demosaicking paper and the McMaster dataset because the images have lower spectral correlations and are comparable to the images taken by color sensors.

B. Implementation Details

We used Pytorch to implement the MARN-JDD network as our framework. We collected 400 images of Panasonic and Canon MSR-Demosaicking for our training

data. We collected 100 images of Panasonic and Canon MSR-Demosaicking [48] for validation.

C. Experimental Setting

We evaluated the proposed deep residual networks for joint demosaicking and denoising with state-of-the-art Kodak data sets and McMaster data sets with 24 and 18 natural images respectively. These test data are displayed again, so that the following experiments differ from the standard camera pipeline used in practical scenarios. By adding white Gaussian noise to the Bayer images, we get the noisy Bayer mosaic input. The sizes of all convolution kernels are 3×3 implementation.

The MARN-JDD system was trained using the stochastic gradient decent with the following parameters.

| | |
|---------------------------|--------------|
| <i>Base learning rate</i> | <i>0.001</i> |
| <i>Channels</i> | <i>3</i> |
| <i>Number of layers</i> | <i>16</i> |
| <i>Batch size</i> | <i>2</i> |
| <i>Training iteration</i> | <i>200</i> |
| <i>Epochs</i> | <i>270</i> |

For optimal results author performed many experiment after that number of factor which improve our result are explain: Learning rate matter a lot to reflect the result to compete with state of arts algorithm so we analysis many training experiment after that we concluded that if we set learning rate to 0.001 after 60 epochs our validation pnsr zig zagging from low to high and high to low so we change learning after 60 epochs to 0.0002 and after 165 epochs to 0.0001. learning rate setting helps us to reduce epochs, increase validation which will affect our testing pnsr. Number of epochs don't affect result after some limit i.e. 270 epochs in our case. In an epoch iteration effect, the result if we increase iteration in an epoch. Our result increase 1-2 dB in iterations increase. Kernel size also effect result as first layer and last layer use 3×3 kernel. Number of layer effect result but also effect the performance in term of time. It's a trade-off between the number of layers and time to demonstrate and denote jointly, so a decision can be made between them. Because if we use more layer for the results, joint demosaicking and denoising takes more milliseconds than more state-of -

the-art algorithms. So, it's a trade-off between the two, so we made a choice at a time when we have little time and more layer. Activation function also improve performance. The latest two activation function are ReLU and PReLU and main argument for the Parametric ReLU over-standard ReLU's is that they do not saturate as you approach the ramp. They don't offer a distinct advantage in most other ways. The effect may not seem dramatic, but in some cases, it can be profoundly advantageous. It also shows that the use of PReLU came at a marginal cost, but we use PReLU. The addition of batch normalization between each convolution and PReLU is destructive for demosaicking, but effective for denoising. It turns out that batch normalization destroys the ability of deep models to generalize in our MARU block, so remove the MARU block and add it to the Denoiser block.

D. Comparison with state-of-the-art algorithms

The results were assessed utilizing the structural similarity (SSIM) index and peak signal-to-noise ratio (PSNR).

We compared the results of ours with equally standard as well as present state-of-the-art algorithms such as FlexISP, ADMM, DeepJoint and DRDD-16J. Figure 5.1 and 5.3 lists the overall performance of all McMaster and Kodak data sets techniques. We can see that MARN JDD performed much better than all the other techniques mentioned for all the pictures in the Kodak, McMaster data set. We have changed the noise level of Bayer input image to different settings: $\sigma = 0$, $\sigma = 5$, $\sigma = 10$, $\sigma = 15$, $\sigma = 20$, and $\sigma = 25$. Note that the results for noise level being zero means the JDD problem degenerates to the original demosaicking problem. MARN-JDD training time is small as compared other state of algorithm. MARN-JDD model has training dataset that contain only 500 images which is too much small as compared to DeepJoint[2] and others.

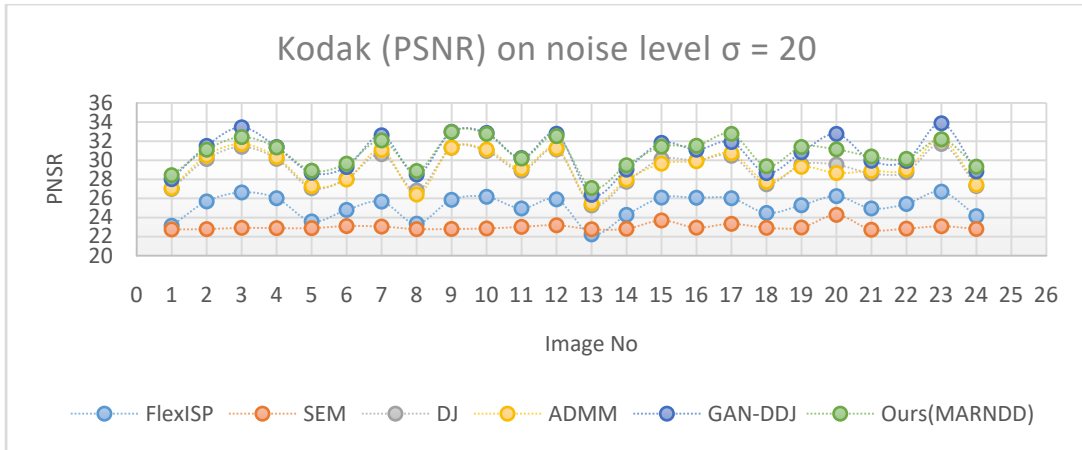


Figure 0.1Kodak per image results (PSNR) on noise level $\sigma = 20$.

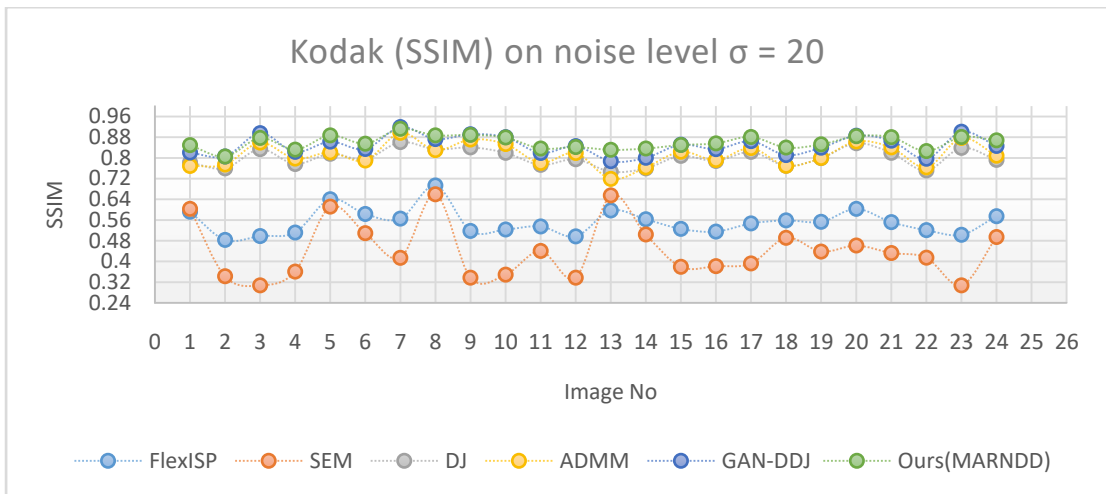


Figure 0.2Kodak per image results (SSIM) on noise level $\sigma = 20$.

Figure 5.1 and 5.2 visually shows Kodak's results in PNSR and SSIM. Figure 5.1 comparison in PNSR show that MARN perform better on 19 images out of 24 images than rest of algorithms. Figure 5.2 comparison in SSIM show that MARN perform better on 21 images out of 24 images than rest of algorithms. FlexISP, DeepJoint and GAN-DDJ results for the Kodak dataset suffer from artifacts such as visible noisy pixels or strange textures caused by denoising. With visible noisy pixels, the results of FlexISP degrade rapidly as the noise level increases. MARNDD's results, however, remain good and stable, showing the strength of the proposed algorithm.

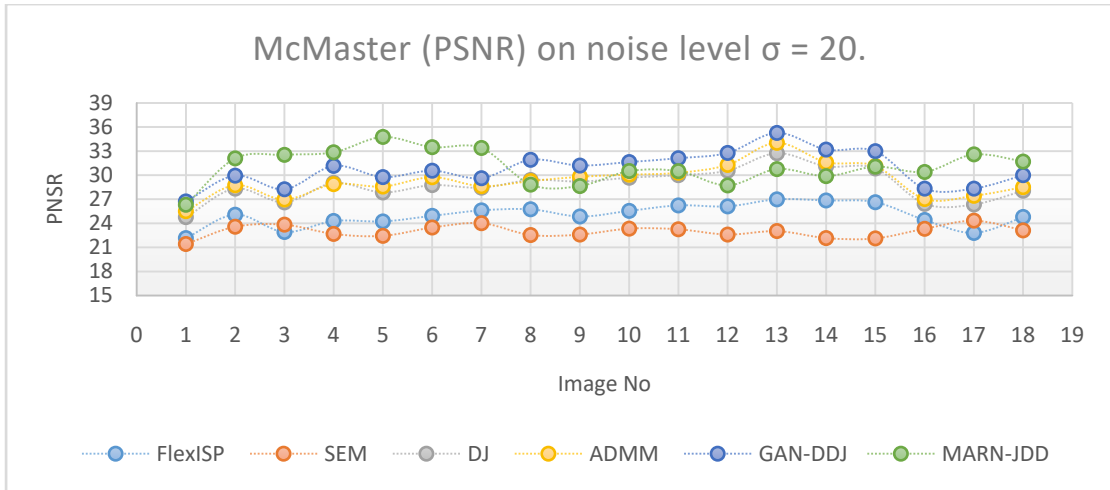


Figure 0.3 McMaster per image results (PSNR) on noise level $\sigma = 20$.

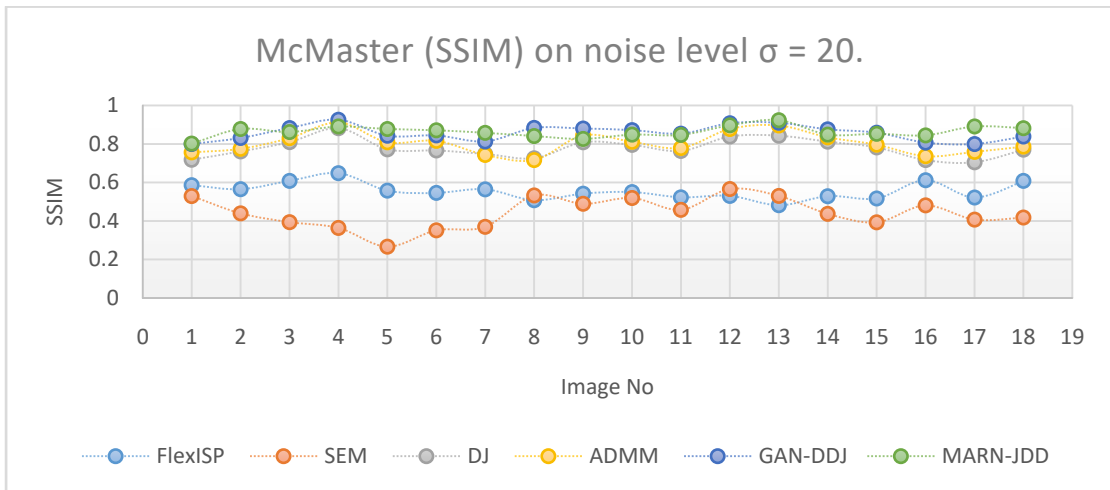


Figure 0.4 McMaster per image results (SSIM) on noise level $\sigma = 20$.

Figure 5.3,5.4 shows the results of PSNR & SSIM on the McMaster dataset with a noise level of $\sigma = 20$. From Figure 5.3 and 5.4 MARN result are stable from state of the arts algorithm due to residually natural of MARN. Figure 5.3 comparison in PNSR show that our MARN perform better on 10 images out of 18 images then rest of algorithms. Figure 5.4 comparison in SSIM show that our MARN perform better on 14 images out of 18 images then rest of algorithms. It can be concluded that in most test cases our method is much better than comparative methods.

Table 4 Average PSNR (dB) comparisons for the joint demosaicking and de-noise results. The best is in bold.

| <i>Dataset</i> | <i>Noise Level</i> | <i>FlexISP</i> | <i>ADMM</i> | <i>DeepJoint</i> | <i>DRDD</i> | <i>MARN-DD</i> |
|--|--------------------|----------------|-------------|------------------|--------------|----------------|
| <i>Kodak</i> (24 <i>images</i>) | $\sigma = 0$ | 34.98 | 31.63 | 39.57 | 40.40 | 38.11 |
| | $\sigma = 5$ | 31.31 | 31.60 | 36.11 | 36.38 | 35.89 |
| | $\sigma = 10$ | 28.64 | 31.04 | 31.65 | N/A | 33.66 |
| | $\sigma = 15$ | 26.67 | 30.16 | 31.28 | 31.59 | 32.07 |
| | $\sigma = 20$ | 25.15 | 29.26 | 29.17 | N/A | 30.87 |
| | $\sigma = 25$ | 23.90 | 28.38 | 26.13 | 29.18 | 29.92 |
| <i>McM</i> (18 <i>images</i>) | $\sigma = 0$ | 35.18 | 32.66 | 37.60 | 37.52 | 34.86 |
| | $\sigma = 5$ | 31.17 | 32.63 | 35.53 | 35.47 | 33.92 |
| | $\sigma = 10$ | 28.51 | 31.72 | 30.95 | N/A | 32.63 |
| | $\sigma = 15$ | 26.55 | 30.50 | 31.28 | 31.49 | 31.51 |
| | $\sigma = 20$ | 25.01 | 29.31 | 28.79 | N/A | 30.55 |
| | $\sigma = 25$ | 23.73 | 28.20 | 26.57 | 28.90 | 29.71 |

We also alternate the noise level of Bayer's input images from $\sigma=$ zero to 25. Results for noise degree $\sigma =0$, which means that the problem degenerates into a demosaicking problem, are additionally listed. The average PSNR results can be found in Table 4. In comparison to DRDD-16J, we achieve an average gain of .59 dB and DeepJoint, we achieve an average gain of 3.64 dB and an average gain of 5.28 dB compared to FlexISP in the Kodak dataset. As for the McMaster dataset, we gain from noisy Bayer input from 2.18 dB to 6.31dB. We can conclude from the table that the higher the noise level, the higher our algorithm outperforms compared to algorithms. In other words, our method is much more robust in noise level variations.

Table 5 Average SSIM comparisons for the joint demosaicking and de-noise results. The best is in bold.

| <i>Dataset</i> | <i>Noise</i> | <i>FlexISP</i> | <i>ADMM</i> | <i>DeepJoint</i> | <i>MARN-</i> |
|----------------|--------------|----------------|-------------|------------------|--------------|
|----------------|--------------|----------------|-------------|------------------|--------------|

| | <i>Level</i> | | | | <i>DD</i> |
|--|---------------|--------|--------|--------|--------------|
| <i>Kodak</i> (24 <i>images</i>) | $\sigma = 0$ | 0.9426 | 0.8873 | 0.9271 | 0.978 |
| | $\sigma = 5$ | 0.8694 | 0.8787 | 0.9058 | 0.951 |
| | $\sigma = 10$ | 0.7583 | 0.8595 | 0.8731 | 0.917 |
| | $\sigma = 15$ | N/A | N/A | N/A | 0.886 |
| | $\sigma = 20$ | 0.5520 | 0.8132 | 0.8015 | 0.858 |
| | $\sigma = 25$ | N/A | N/A | N/A | 0.831 |
| <i>McM</i> (18 <i>images</i>) | $\sigma = 0$ | 0.9385 | 0.9052 | 0.8876 | 0.937 |
| | $\sigma = 5$ | 0.8627 | 0.8966 | 0.8739 | 0.921 |
| | $\sigma = 10$ | 0.7534 | 0.8699 | 0.8467 | 0.899 |
| | $\sigma = 15$ | N/A | N/A | N/A | 0.876 |
| | $\sigma = 20$ | 0.5556 | 0.8046 | 0.7789 | 0.853 |
| | $\sigma = 25$ | N/A | N/A | N/A | 0.830 |

The average SSIM results can be found in Table 5. In comparison to DeepJoint, we achieve an average gain of .05 and ADMM, we achieve an average gain of .04 and an average gain of .09 compared to FlexISP in the Kodak dataset.

The subjective quality comparison results are shown in Figs.12-18 with a noise level of $\sigma = 20$. We note that the SEM method suffers from a lack from of noise robustness; FlexISP, DRDD and DeepJoint methods suffers from various artifacts such as vertical color lines, remaining noisy pixels and unnatural color. The ADMM algorithm is relatively good among the competing approaches, but in terms of visual quality still falls behind when compared to our method. In terms of fewer artifacts, better preserved fine details (e.g. flower petals, wood texture patterns and hairs) and more vivid color showing the superiority and robustness of the proposed algorithm, the reconstructed images by our method appear much better visually.

Table 6 Running Time of Joint Demosaicking and Denoising (seconds)

| Dataset (Image Size) | FlexISP (CPU) | ADMM (CPU) | DeepJoint (CPU/GPU) | DRDD-16 (CPU/GPU) | MARN(GPU) |
|----------------------|---------------|------------|---------------------|-------------------|-----------|
| McM (500 × 500) | 233.94 | 705.47 | 5.43 / 0.12 | 3.93 / 0.88 | 0.73 |

E. Running Time Comparison

Table 5 provides an average comparison of joint demo and denoising runtime. The data are measured on a Google Colab and other documents. With proper settings, all methods process 500× 500 images only once without cutting or scaling, which ensures

a fair comparison. DeepJoint, MARN are much faster than FlexISP and ADMM. MARN is marginal fast from DRDD.

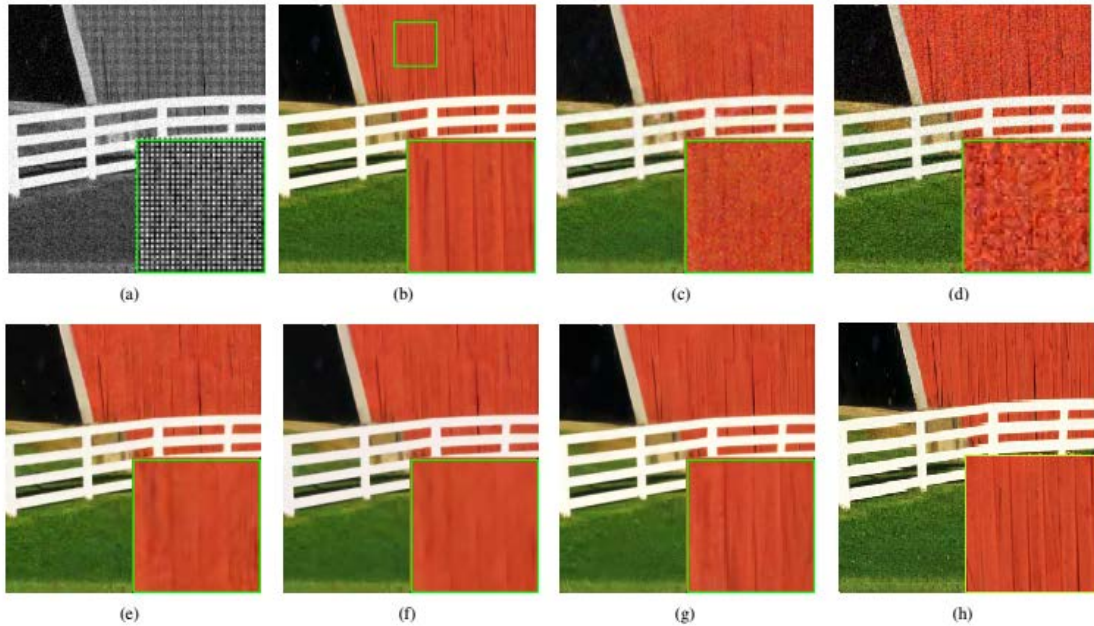


Figure 0.5 Visual Results of McMaster18 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result (PSNR=24.80, SSIM=0.6084); (d) SEM result (PSNR=23.11, SSIM=0.4183); (e) DeepJoint result (PSNR=28.07, SSIM=0.7706); (f) ADMM result (PSNR=28.50, SSIM=0.7862); (g) GAN result (PSNR=30.00, SSIM=0.8387); (i) our MARN-JDD result (PSNR=30.164, SSIM=0.842)

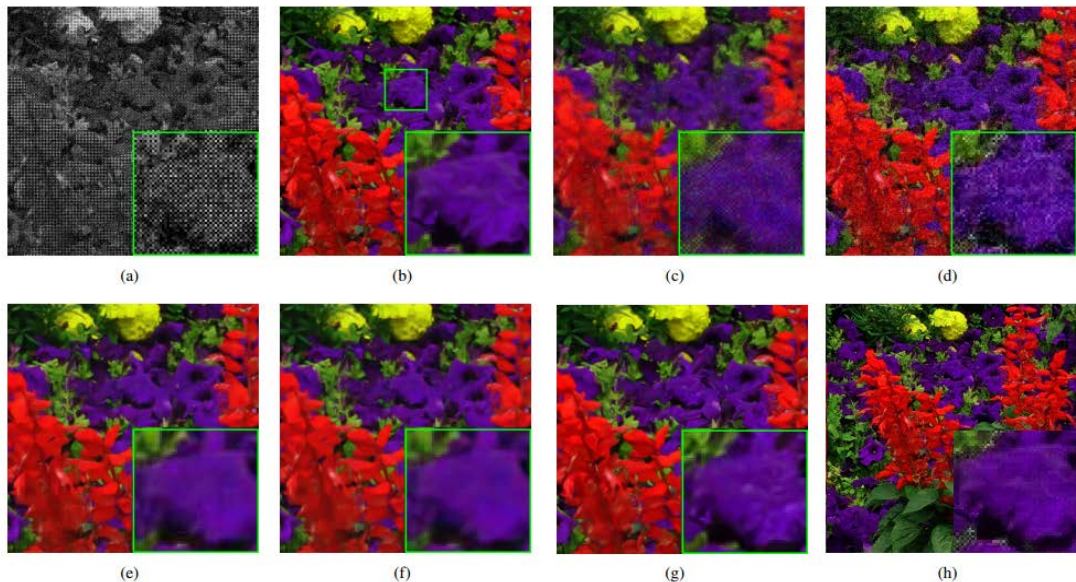


Figure 0.6 Visual Results of McMaster17 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result (PSNR=22.80, SSIM=0.5217); (d) SEM result (PSNR=24.33, SSIM=0.4072); (e) DeepJoint result (PSNR=26.35, SSIM=0.7051); (f) ADMM result (PSNR=27.42,

$SSIM=0.7586$); (g) GAN result ($PSNR=28.33$, $SSIM=0.8000$) (h) our MARN-JDD result ($PSNR=27.66$, $SSIM=0.805$)

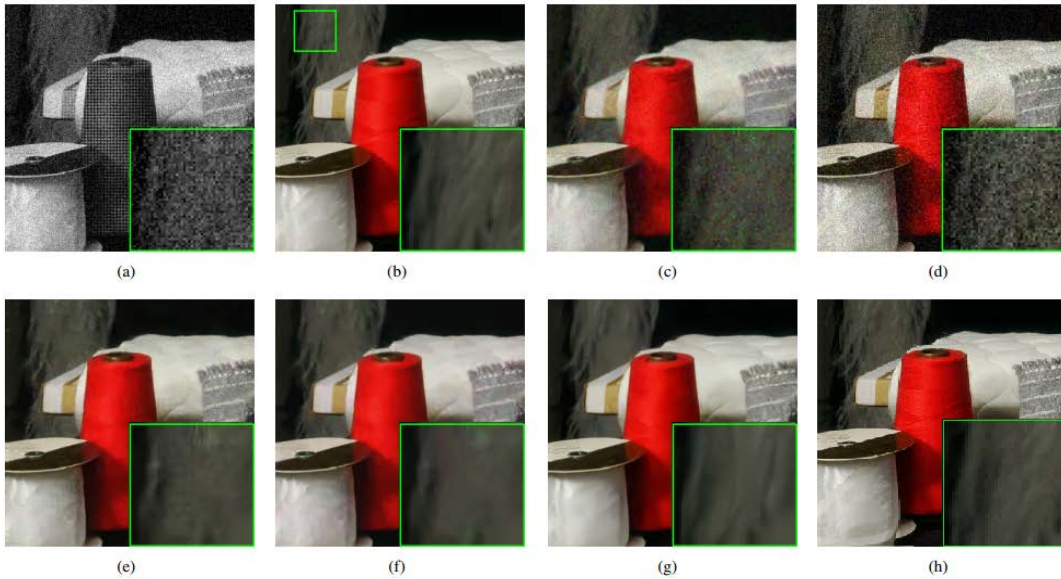


Figure 0.7 Visual Results of McMaster7 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result ($PSNR=25.62$, $SSIM=0.5647$); (d) SEM result ($PSNR=24.02$, $SSIM=0.3696$); (e) DeepJoint result ($PSNR=28.44$, $SSIM=0.7478$); (f) ADMM result ($PSNR=28.62$, $SSIM=0.7434$); (g) GAN result ($PSNR=29.63$, $SSIM=0.8134$); (h) our MARN-JDD result ($PSNR=30.31$, $SSIM=0.837$).

Chapter 6

CONCLUSION

This thesis determines that quality of demosaicking and denoising is always enhance by a joint strategy according to a deep neural. MARN-JDD network turnout excellent results equally qualitative and quantitative when compared with the present state-of-the-art network. While doing so, MARN JDD strategy is able to generalize well even if it is trained on small data sets and the number of our network variables is low compared to other competing solutions. Experimental results show that in objective and subjective evaluations, the proposed method outperforms several existing or perhaps state-of - the-art techniques.

Chapter 7

Bibliography

- [1] Xingyu ZHANG*, Ming-Ting SUN†, Lu FANG‡ and Oscar C. AU, “JOINT DENOISING AND DEMOSAICKING OF NOISY CFA IMAGES BASED ON INTER-COLOR CORRELATION” in 2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)
- [2] `Gharbi, M., Chaurasia, G., Paris, S., Durand, F. 2016. Deep Joint Demosaicking and Denoising.
- [3] S. Zhang, X. Li and C. Zhang, "Modified Adaptive Median Filtering," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, 2018, pp. 262-265.

- [4] J. Mizutani, S. Ogawa, K. Shinoda, M. Hasegawa and S. Kato, "Multispectral demosaicking algorithm based on inter-channel correlation," 2014 IEEE Visual Communications and Image Processing Conference, Valletta, 2014, pp. 474-477.
- [5] N. Li, J. S. J. Li, S. Randhawa and D. G. Bailey, "Edge preserving CFA demosaicking based on nonlinear weighted color differences," 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 1143-1146.
- [6] S. P. Jaiswal, L. Fang, V. Jakhetiya, J. Pang, K. Mueller and O. C. Au, "Adaptive Multispectral Demosaicking Based on Frequency-Domain Analysis of Spectral Correlation," in IEEE Transactions on Image Processing, vol. 26, no. 2, pp. 953-968, Feb. 2017
- [7] M. J. Ehrhardt and S. R. Arridge, "Vector-Valued Image Processing by Parallel Level Sets," in IEEE Transactions on Image Processing, vol. 23, no. 1, pp. 9-18, Jan. 2014.
- [8] Wang, Y.-Q. (2015). A multilayer neural network for image demosaicking. 2014 IEEE International Conference on Image Processing, ICIP 2014. 1852-1856
- [9] Kapah, Oren & Hel-Or, Hagit. (2000). Demosaicking using artificial neural networks. Proceedings of the SPIE – Applications of Artificial Neural Networks in Image Processing V. 3962. 112-120.
- [10] Wu, Xiaolin. (2011). Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic Imaging.
- [11] K. Hirakawa and T. W. Parks (2006). Joint demosaicing and denoising, IEEE Transactions on Image Processing, V. 15. 2146–2157.
- [12] H. S. Malvar, L. He, and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, May 17-21, 2004, pp. 485–488.
- [13] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," IEEE Trans. Image Processing, vol. 14, no. 12, pp. 2167–2178, 2005.

- [14] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in Proceedings of the International Conference on Image Processing, ICIP, September 26-29, 2010, pp. 137–140.
- [15] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electronic Imaging*, vol. 20, no. 2, p. 023016, 2011.
- [16] F. He, Y. F. Wang, and K. Hua, "Self-learning approach to color demosaicking via support vector regression," in 19th IEEE International Conference on Image Processing, ICIP, September 30 - October 3, 2012, pp. 2765–2768.
- [17] Y. M. Lu, M. Karzand, and M. Vetterli, "Demosaicking by alternating projections: Theory and fast one-step implementation," *IEEE Trans. Image Processing*, vol. 19, no. 8, pp. 2085–2098, 2010.
- [18] Y. Wang, "A multilayer neural network for image demosaicking," in IEEE International Conference on Image Processing, ICIP, October 27-30, 2014, pp. 1852–1856. [9] J. Duran and A. Buades, "A demosaicking algorithm with adaptive interchannel correlation," *IPOL Journal*, vol. 5, pp. 311–327, 2015.
- [19] D.Kiku, Y.Monno, M.Tanaka, and M.Okutomi, "Residual interpolation for color image demosaicking," in IEEE International Conference on Image Processing, ICIP, September 15-18, 2013, pp. 2304–2308.
- [20] Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color image demosaicking," in IEEE International Conference on Image Processing, ICIP, September 27-30, 2015, pp. 3861–3865.
- [21] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-laplacian residual interpolation for color image demosaicking," in Digital Photography X, part of the IS&T-SPIE Electronic Imaging Symposium, San Francisco, California, USA, February 2, Proceedings., 2014, p. 90230L.
- [22] "Beyond color difference: Residual interpolation for color image demosaicking," *IEEE Trans. Image Processing*, vol. 25, no. 3, pp. 1288–1300, 2016.

- [23] W. Ye and K. Ma, “Color image demosaicing using iterative residual interpolation,” *IEEE Trans. Image Processing*, vol. 24, no. 12, pp. 5879–5891, 2015.
- [24] K. Hua, S. C. Hidayati, F. He, C. Wei, and Y. F. Wang, “Context aware joint dictionary learning for color image demosaicking,” *J. Visual Communication and Image Representation*, vol. 38, pp. 230–245, 2016.
- [25] J. Go, K. Sohn, and C. Lee, “Interpolation using neural networks for digital still cameras,” *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 610–616, 2000.
- [26] H. Z. Helor, “Demosaicking using artificial neural networks,” *Proc Spie*, vol. 45, no. 3962, pp. 112–120, 2000.
- [27] F. L. He, Y. C. F. Wang, and K. L. Hua, “Self-learning approach to color demosaicking via support vector regression,” in *IEEE International Conference on Image Processing*, 2013, pp. 2765–2768.
- [28] J. Sun and M. F. Tappen, “Separable markov random field model and its applications in low level vision,” *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 22, no. 1, p. 402, 2013.
- [29] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Computer Science*, 2014.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [32] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.
- [33] C. Ledig, Z. Wang, W. Shi, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, and A. Tejani, “Photo-realistic single image super-resolution using a generative adversarial network,” pp. 105– 114, 2016.

- [34] 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.
- [35] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *Acm Transactions on Graphics*, vol. 35, no. 6, p. 191, 2016.
- [36] R. Tan, K. Zhang, W. Zuo, and L. Zhang, "Color image demosaicking via deep residual learning," in *IEEE International Conference on Multimedia and Expo*, 2017, pp. 793–798
- [37] Zhang et al., "Adaptive Residual Networks for High-Quality Image Restoration," in *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3150-3163, July 2018.
- [38] W. Yang et al., "Deep Edge Guided Recurrent Residual Learning for Image Super-Resolution," in *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5895-5907, Dec. 2017.
- [39] D. Kiku, Y. Monno, M. Tanaka and M. Okutomi, "Beyond Color Difference: Residual Interpolation for Color Image Demosaicking," in *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1288-1300, March 2016.
- [40] Y. Kim and J. Jeong, "Four-Direction Residual Interpolation for Demosaicking," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 881-890, May 2016.
- [41] Maas, A.L., Hannun, A.Y., Ng, A.Y. "Rectifier nonlinearities improve neural network acoustic models", in: *Proc. ICML*. 2013
- [42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, , pp. 1–14 April 2015
- [43] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, pp. 1026–1034, 2015
- [44] DeepDemaicking: Adaptive Image Demosaicking via Multiple Deep Fully Convolutional Networks," in *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2408-2419, May 2018.

- [45] Ruofan Zhou and Radhakrishna Achanta and Sabine strunk,” Deep Residual Network for Joint Demosaicing and Super-Resolution,” in ArXiv Vol. 1802.06573, 2018
- [46] Kokkinos, Filippos and Lefkimmiatis, Stamatias,” Iterative Residual Network for Deep Joint Image Demosaicking and Denoising,” in ArXiv vol.1807.06403 July 2018
- [47] Tan, Hanlin& Xiao, Huaxin& Lai, Shiming& Liu, Yu & Zhang, Maojun. Deep Residual Learning for Image Demosaicing and Blind Denoising. 10.13140/RG.2.2.11048.62725. 2018
- [48] Khashabi, D., Nowozin, S., Jancsary, J., Fitzgibbon, A.W.: Joint demosaicking and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing* 23(12) (Dec 2014)
- [49] B. E. Bayer, “Color imaging array,” July 20 1976. US Patent 3,971,065.
- [50] X. Li, B. Gunturk, and L. Zhang, “Image demosaicing: A systematic survey,” in *Electronic Imaging 2008*, pp. 68221J–68221J, International Society for Optics and Photonics, 2008
- [51] H. S. Malvar, L.-w. He, and R. Cutler, “High quality linear interpolation for demosaicing of bayer-patterned color images,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP’04). IEEE International Conference on*, vol. 3, pp. iii–485, IEEE, 2004.
- [52] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5,no. 4, pp. 115–133, 1943
- [53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097– 1105, 2012.
- [54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet,S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke,and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [55] N. Srivastava, G. E. Hinton, A. Krizhevsky,I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from

overfitting.,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

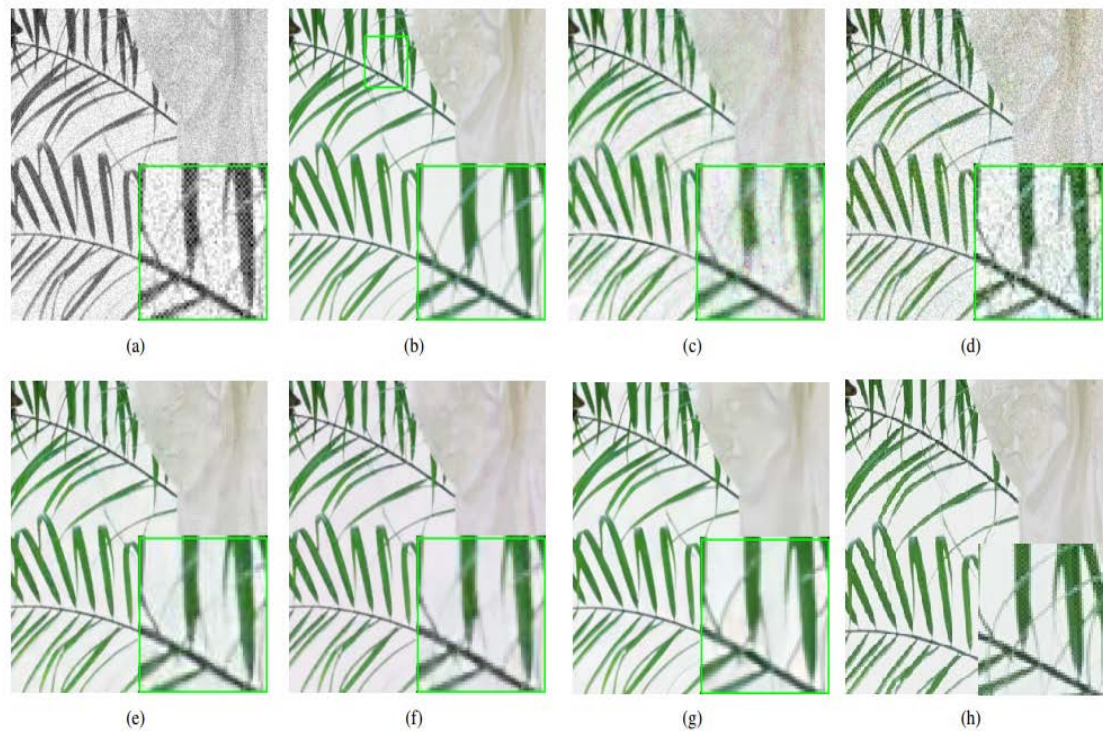


Figure 0.1 Visual Results of McMaster4 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result(PSNR=24.30, SSIM=0.6481); (d) SEM result(PSNR=22.67, SSIM=0.3639); (e) DeepJoint result(PSNR=29.04, SSIM=0.8819); (f) ADMM result(PSNR=28.89, SSIM=0.9119); (g) GAN result(PSNR=31.17, SSIM=0.9261); (h) our MARN-JDD result(PSNR=30.50, SSIM=0.92).

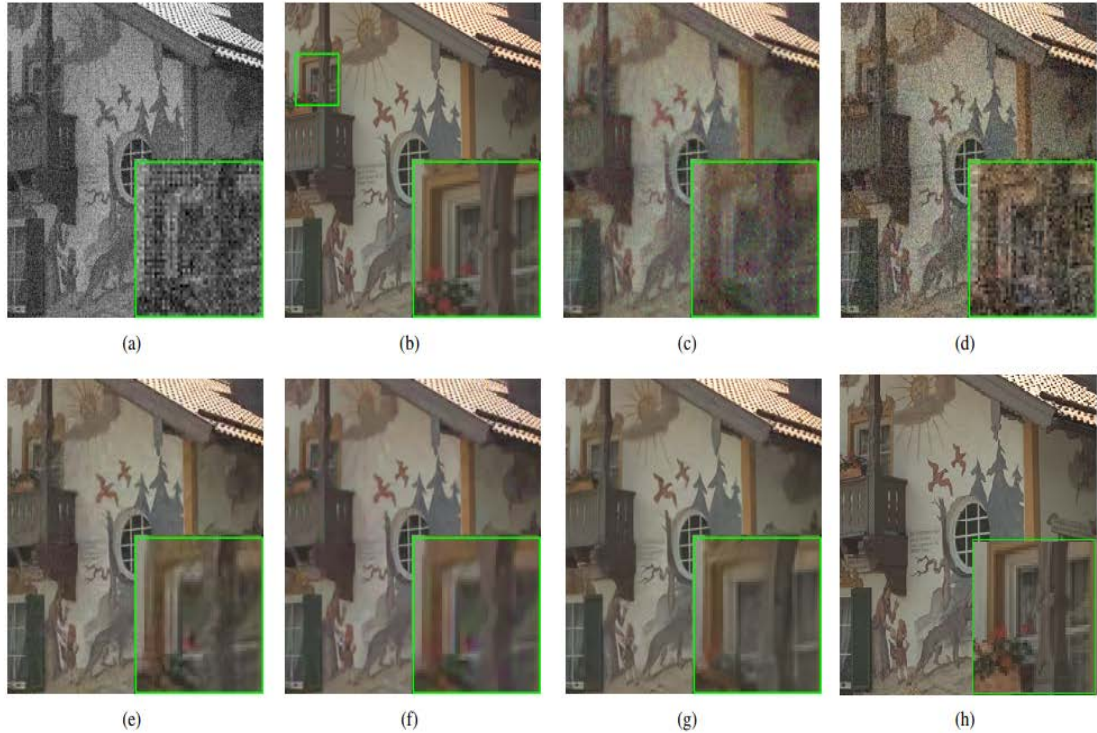


Figure 0.2 Visual Results of kodak24 with $\sigma = 20$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result (PSNR=24.14, SSIM=0.5745); (d) SEM result (PSNR=22.79, SSIM=0.4939); (e) DeepJoint result (PSNR=27.30, SSIM=0.7925); (f) ADMM result (PSNR=27.43, SSIM=0.8082); (g) our GAN result (PSNR=28.82, SSIM=0.8460); (h) our MARN-JDD result (PSNR=29.32, SSIM=0.86).

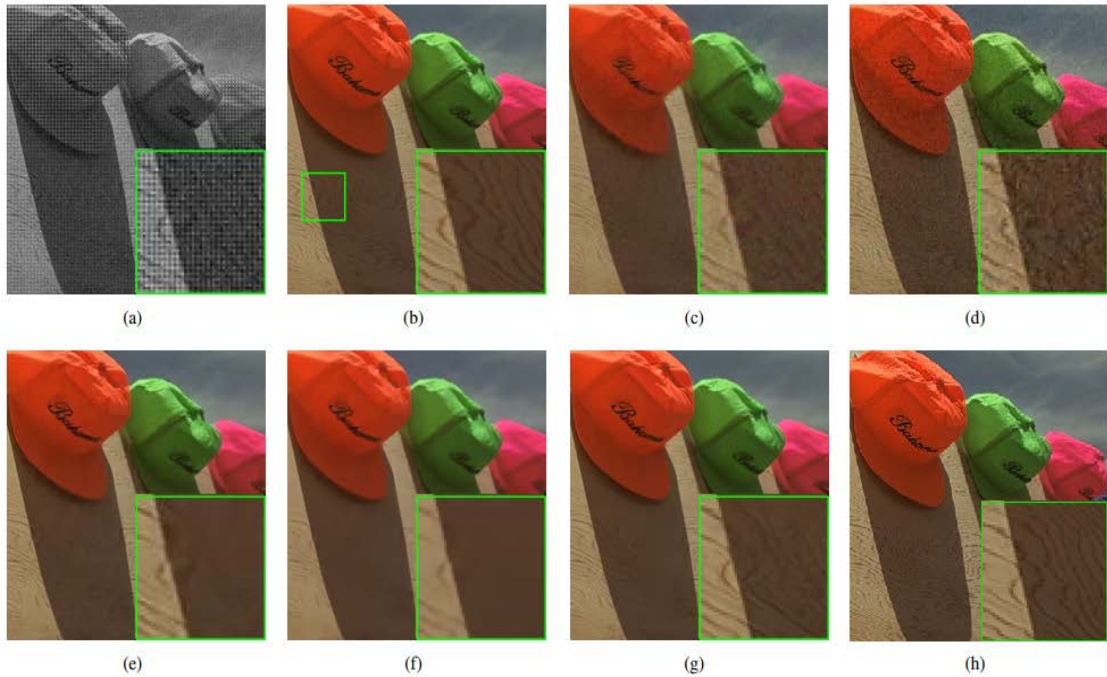


Figure 0.3 Visual Results of kodak3 with $\sigma = 10$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result (PSNR=30.90, SSIM=0.7521); (d) SEM result (PSNR=30.36, SSIM=0.6973); (e) DeepJoint result (PSNR=33.99, SSIM=0.9009); (f) ADMM result

(PSNR=33.40, SSIM=0.8949); (g) GAN result(PSNR=36.57, SSIM=0.9370);
 (h) our MARN-JDD result(PSNR=35.28, SSIM=0.927).

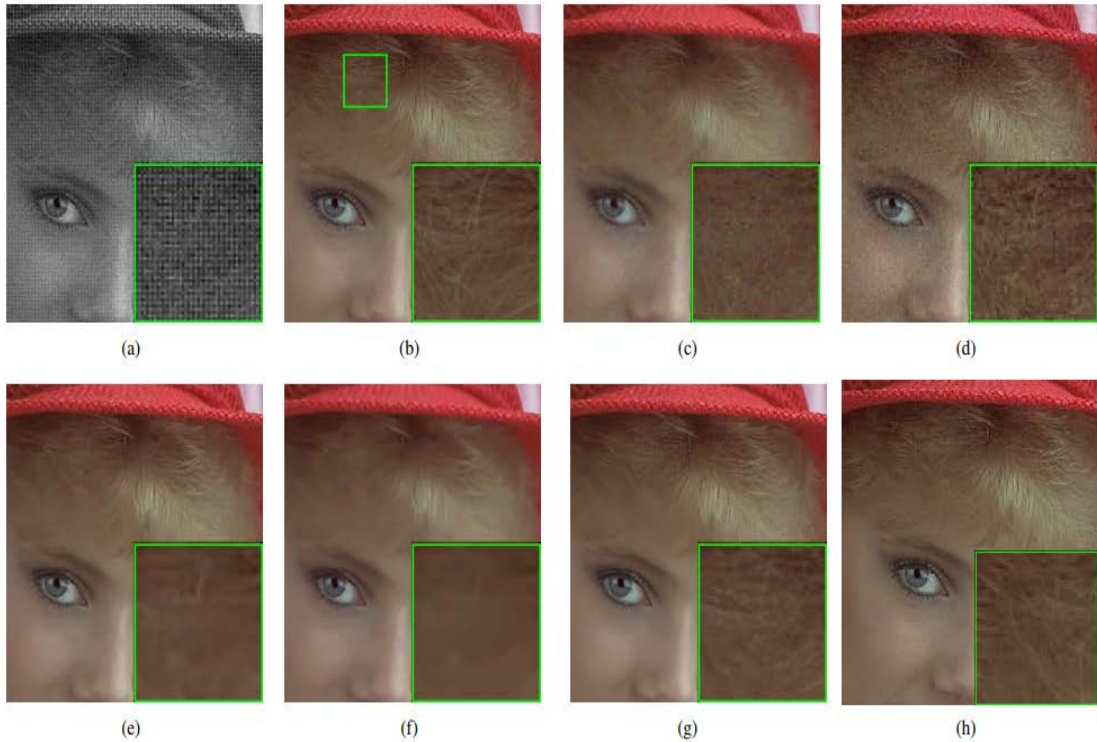


Figure 0.4 Visual Results of kodak4 with $\sigma = 10$ noise for Joint denoising and demosaicking. (a) BayerNoisy image;(b) Original image; (c) FlexISP result (PSNR=29.67, SSIM=0.7395); (d) SEM result (PSNR=29.63, SSIM=0.7055);(e) DeepJoint result (PSNR=32.43, SSIM=0.8495); (f) ADMM result (PSNR=31.93, SSIM=0.8414); (g) GAN result (PSNR=34.27, SSIM=0.8928). (h) our MARN-JDD result (PSNR=34.45, SSIM=0.898).

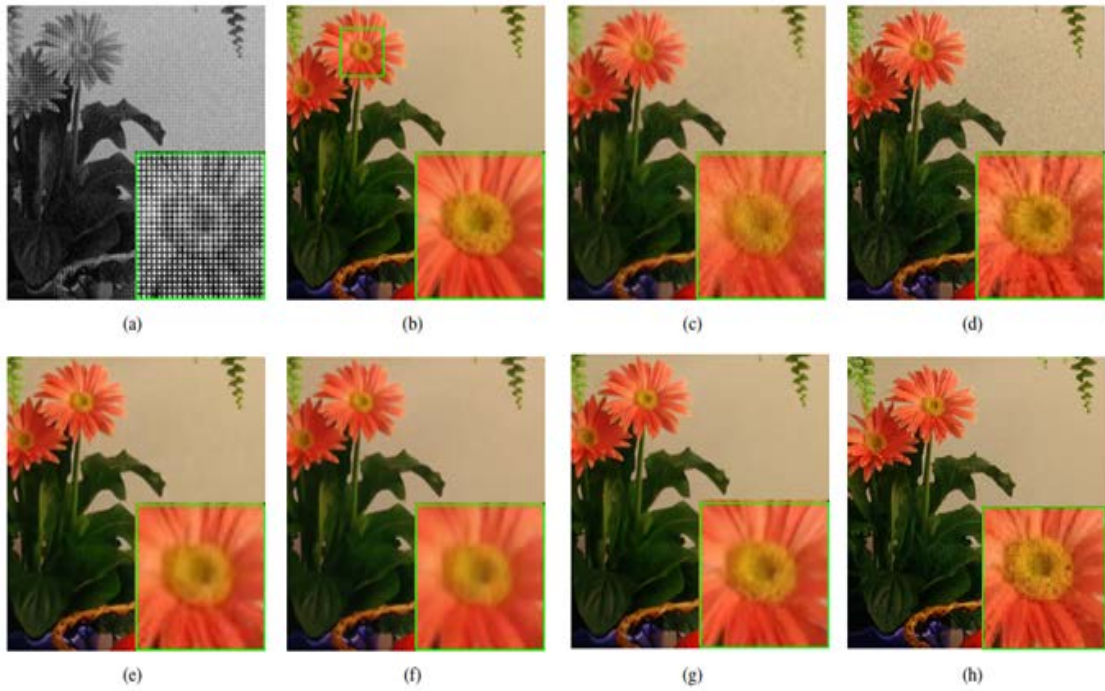


Figure 0.5 Visual Results of kodak9 with $\sigma = 10$ noise for Joint denoising and demosaicking. (a) BayerNoisy image; (b) Original image; (c) FlexISP result(PSNR=30.53, SSIM=0.7621); (d) SEM result(PSNR=30.71, SSIM=0.7244); (e) DeepJoint result(PSNR=34.01, SSIM=0.9031); (f) ADMM result(PSNR=32.99, SSIM=0.9025); (g) GAN result(PSNR=36.05, SSIM=0.9280); (h) our MARN-JDD result(PSNR=36.04, SSIM=0.925)