BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

PHOTO NOISE REDUCTION USING DEEP NEURAL NETWORKS

REDUKCE ŠUMU VE FOTOGRAFIÍCH POMOCÍ HLUBOKÝCH NEURONOVÝCH SÍTÍ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE
JONÁŠ TICHÝ

SUPERVISOR
VEDOUCÍ PRÁCE
Ing. MICHAL ŠPANĚL, Ph.D.

BRNO 2022

Department of Computer Graphics and Multimedia (DCGM) Academic year 2021/2022

Bachelor's Thesis Specification

25132

Student:	**Tichý Jonáš**
Programme:	Information Technology
Title:	**Photo Noise Reduction Using Deep Neural Networks**
Category:	Image Processing

Assignment:

1. Get familiar with deep neural networks and their learning.
2. Get acquainted with the problem of image denoising and current methods based on deep neural networks.
3. Prepare a dataset for your own experiments.
4. Implement chosen digital photo denoising methods and experiment with them.
5. Evaluate and compare your results using appropriate metrics like PSNR or SSIM. Discuss possible future work.
6. Create a short poster or video presenting your work, its goals and results.

Recommended literature:

- Tian *et al.*, "Deep Learning on Image Denoising: An overview", Elsevier, Neural Networks, 2020 (https://doi.org/10.1016/j.neunet.2020.07.025).
- Yang *et al.*, "Deep Learning for Single Image Super-Resolution: A Brief Review", IEEE Transactions on Multimedia, 2019 (https://dl.acm.org/doi/10.1109/TMM.2019.2919431).

Requirements for the first semester:

- The first three items of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Španěl Michal, Ing., Ph.D.**
Head of Department:	Černocký Jan, doc. Dr. Ing.
Beginning of work:	November 1, 2021
Submission deadline:	May 11, 2022
Approval date:	November 2, 2021

Abstract

Image noise is a fundamental problem in digital photography. The goal of this thesis is to study the use of deep neural networks in denoising of digital photographs. Two different denoising methods based on deep neural networks, DnCNN and BRDNet, were implemented and their performance was measured in several experiments. Additionally, a user testing experiment was designed and carried out to evaluate the perceived image quality of the studied methods by the general public. The experiments have shown that while both methods achieve state-of-the-art denoising results in metrics such as PSNR and SSIM, the perceived visual quality does not always correlate with the numerical metrics. The results presented in this thesis highlight the importance of proper training datasets and image quality metrics in digital photography denoising.

Abstrakt

Obrazový šum je fundamentálním problémem v digitální fotografii. Cílem této práce je studium redukce šumu ve fotografiích pomocí hlubokých neuronových sítí. Dvě vybrané metody založené na hlubokých neuronových sítích, DnCNN a BRDNet, byly implementovány a jejich výkon byl změřen v několika experimentech. Kromě toho byl navržen a proveden experiment na uživatelích s cílem vyhodnotit vnímanou kvalitu obrazu širokou veřejností. Experimenty ukázaly, že zatímco obě metody dosahují výborných výsledků v metrikách, jako je PSNR a SSIM, vnímaná vizuální kvalita ne vždy koreluje s numerickými metrikami. Výsledky prezentované v této práci zdůrazňují důležitost vhodných trénovacích dat a metrik kvality obrazu v odšumování digitálních fotografií.

Keywords

denoising, digital photography, deep neural networks, convolutional neural networks, DnCNN, BRDNet, perceived visual quality, user testing

Klíčová slova

odstraňování šumu, digitální fotografie, hluboké neuronové sítě, konvoluční neuronové sítě, DnCNN, BRDNet, vnímaná vizuální kvalita, uživatelské testování

Reference

TICHÝ, Jonáš. *Photo Noise Reduction Using Deep Neural Networks*. Brno, 2022. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Michal Španěl, Ph.D.

Rozšířený abstrakt

Šum v digitální fotografii je jedním z hlavních prvků, které snižují kvalitu výsledného obrazu. S rostoucí popularitou digitální fotografie je čím dál tím větší zájem o digitální odstraňování tohoto nežádoucího prvku. V minulosti bylo pro tento účel hojně využíváno algoritmických metod založených na filtrech. Tyto přístupy mají však mnoho nedostatků, jako je například potřeba manuálního nastavování parametrů, které je činí nedostatečnými pro potřeby digitální fotografie. V posledním desetiletí se v problematice odstraňování šumu začalo hojně využívat metod založených na hlubokých neuronových sítích, které vykazují slibné výsledky.

Cílem této bakalářské práce je seznámit se s využitím metod založených na hlubokých neuronových sítích v oblasti odstraňování šumu a zhodnotit vhodnost jejich využití v digitální fotografii. Pro tyto účely byly vybrány dvě metody, DnCNN a BRDNet, které slibují úctyhodné výsledky na prezentovaných datech. Pro účely srovnání pokroků v tomto odvětví byly tyto metody testovány v několika experimentech za použití vhodných veřejně dostupných dat. Během těchto experimentů byla dokázána důležitost správných trénovacích dat pro využití v odstraňování šumu z fotografií. Metody trénované na fotografiích vykazovaly zlepšení o zhruba 3,1 dB PSNR a 0,11 SSIM nad metodami trénovanými na datech se syntetickým obrazovým šumem. Následně byly také výsledky obou metod porovnány mezi sebou pro ustanovení vlivu rozdílů v architektuře a věku metody na výsledný výkon v odstraňování šumu. Metoda BRDNet podávala konzistentně lepší výsledky než starší metoda DnCNN hlavně díky její širší architektuře a díky tomu lepší schopnosti zachovat jemné detaily ve fotografii. Díky výsledkům z testování bylo také poukázáno na určité nesrovnalosti v původních publikacích těchto metod.

Další zkoumanou oblastí byla vhodnost současně používaných metrik, jako je PSNR nebo SSIM, pro využití k hodnocení kvality obrazu v této problematice. Pro účely srovnání těchto metrik s vnímanou vizuální kvalitou u lidí byl navrhnut a proveden uživatelský experiment. Uživatelé měli v tomto experimentu za úkol srovnávat relativní vizuální kvalitu výstupů obou studovaných metod a původní zašuměné fotografie. Výsledky z tohoto testu ukázaly, že na vnímanou vizuální kvalitu výstupů obou metod má velký vliv míra obrazového šumu. U některých případů s nízkou úrovní obrazového šumu korespondenti označili výstup obou metod jako horší než původní fotografie. Díky těmto shledáním byla vytvořena hypotéza týkající se vlivu úrovně obrazového šumu na výkon odšumovacích metod, která byla následně statisticky potvrzena na požadované hladině významnosti. Výsledky z tohoto testu také odhalily, že ačkoliv metoda BRDNet vykazovala v metrikách PSNR a SSIM jasnou převahu nad metodou DnCNN, uživatelé mezi nimi neshledali žádné statisticky signifikantní rozdíly ve vnímané vizuální kvalitě.

Výsledky z této práce demonstrovaly schopnosti a pokroky současných metod založených na hlubokých neuronových sítích v problematice odstraňování šumu v digitálních fotografiích. Díky extenzivním experimentům bylo také možno poukázat na slabou korelaci tradičních metrik pro určení kvality obrazu se skutečnou vnímanou kvalitou obrazu. Tyto metriky jsou avšak stále hojně používané v nových publikacích a tento problém zůstává většinově bez povšimnutí. Bylo také poukázáno na problémy současných metod v odstraňování šumu z fotografií s nižšími úrovněmi šumu. Z nasbíraných poznatků byly vyvozeny vhodné závěry a byly navrženy různé možnosti navazující práce v této problematice.

Photo Noise Reduction Using Deep Neural Networks

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Michal Španěl, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Jonáš Tichý
May 11, 2022

Acknowledgements

I would like to thank my supervisor Ing. Michal Španělm Ph.D. for his guidance and helpful suggestions. I would also want to thank all individuals that took their time to participate in the public survey.

Contents

Chapter 1

Introduction

With the continuously growing popularity of social media and messaging applications, the number of digital photos taken every day is rapidly increasing. While the hardware and with it the quality of the taken photographs are improving constantly, the demand for small-scale imaging devices, such as smartphones makes it challenging to capture high-quality images in low-light conditions. In turn, there is an urgent need to digitally remove the image noise from such images while preserving the main characteristics of a realistic photograph.

The problem of image noise reduction is hardly new and has been extensively studied for decades. The problem on hand is not concerning only digital photography but extends to fields such as medical or astronomical imagery which further increases the demand for new and improving denoising methods. Early algorithmic approaches for solving the noise reduction task showed moderate success but suffered from several drawbacks that made their usefulness limited. In the past decade with the advancements both in hardware and software, deep neural networks have seen a surge in popularity for various image restoration tasks including image denoising, which show excellent and promising results.

This thesis aims to explore the advancements made in the field of noise reduction in real photographs by deep neural networks. Several experiments are carried out on the implemented studied methods to compare their performance and evaluate their fitness for use in digital photography. Additionally, to address the shortcomings of purely numerical evaluation, a user testing experiment is presented which aims to evaluate the perceived visual quality of the studied methods on the general public. Data from this experiment are presented and used to evaluate the correlation between numerical and human testing results.

Firstly, this thesis introduces the basic principles of image noise in digital photography in Chapter 2 along with an overview of traditionally used noise reduction techniques and commonly used evaluation metrics used to numerically judge the image quality. Chapter 3 then presents currently used methods for image noise reduction based on deep neural networks. Commonly used datasets for various denoising tasks are also presented later in the chapter. The proposed solution for noise reduction is presented in Chapter 4, with the design of proposed testing on real users presented in Chapter 5. Implementation details along with used tools and technologies are described in detail in Chapter 6. Lastly, Chapter 7 presents the results of all experiments that were carried out along with extensive data interpretation.

Chapter 2

Noise reduction in digital images

This chapter provides insight into the conception and structure of different types of image noise present in digital photography. The history of traditional denoising approaches is then introduced along with their advantages and shortcomings. Finally, metrics that are commonly used to evaluate image quality are presented and compared.

2.1 Noise sources in digital images

A digital image is generated by converting the light coming from a natural scene to numerical pixel values. During this process, multiple factors can affect the image signal and lead to what is known as image noise.

When an image sensor is exposed to light in order to capture an image, the average incident energy of photons that land on the said sensor can be seen as virtually constant over a longer period of time. However, over shorter periods the amount of photons incident on the camera sensor can fluctuate. Furthermore, not all photons are perfectly converted to electric charge due to inaccuracies in the hardware technology. This phenomenon is collectively known as *shot noise*. An important feature of this type of noise is the fact that it is *signal-dependent*, meaning that the electric charge fluctuations over time are proportional to the photon flux. This means that different parts of the captured image can be affected by different amounts of shot noise [3]. This type of noise is modeled by the family of Poisson distributions.

Another type of noise that commonly corrupts digital images is *thermal noise* which is generated by a phenomenon called thermal agitation. This phenomenon causes the semiconductors that are used in the camera sensors to randomly emit charges due to heat even without any external electric potential to stimulate them. This type of noise is proportional to the working temperature of the sensor – the higher the temperature, the stronger the effect. In some applications, such as optical astronomy, this effect is minimized by cooling the imaging sensor. This technique is however not feasible for consumer-grade cameras and thus this type of noise remains an issue. This type of noise is *signal-independent* and is suitably modeled by Gaussian distribution [12].

Furthermore, during the process of capturing a digital image, there is a possibility of correlation in charges of neighboring sensor elements. In other words, the electric charge recovered from a sensor element does not have to be influenced only by the incident photons but also by charges accumulated by the surrounding sensor elements. This is commonly referred to as *cross-talk* and it is usually modeled by the adoption of correlated noise.

Lastly, the electric potential from the image sensor is usually amplified and converted by the analog-to-digital-converter. This process may introduce further noise and/or strengthen the existing noise. The image signal processing (ISP) pipeline is then responsible for image postprocessing. The ISP pipeline can further alter the properties of the captured image by performing exposure compensation, white balance, color space conversion, gamma correction, and more depending on the manufacturer and camera settings [16].

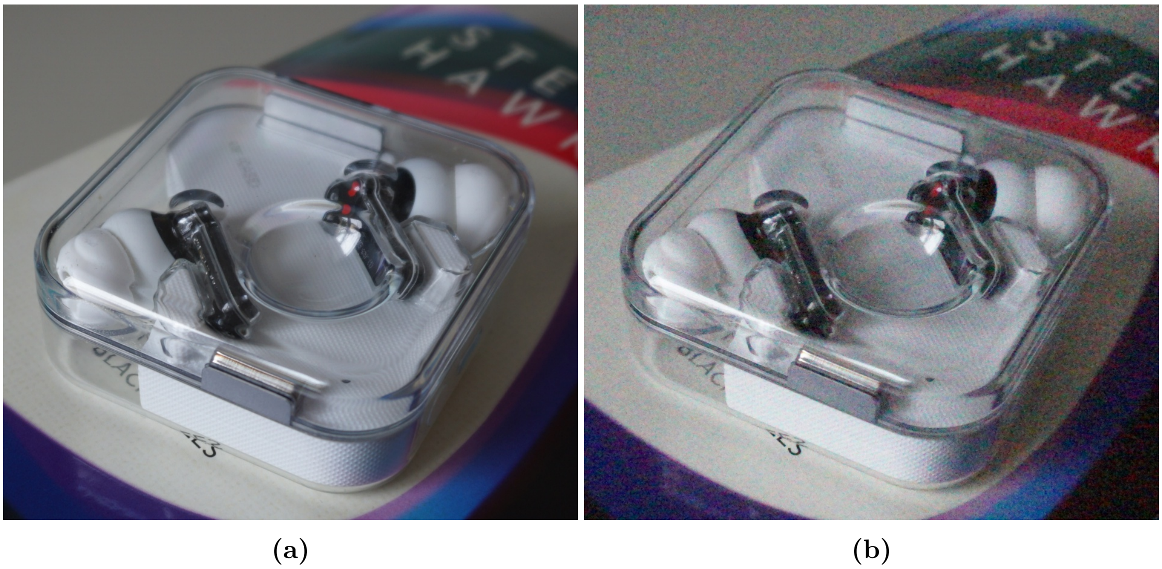


Figure 2.1: Two images demonstrating the effect of low and high levels of image noise. Both images were captured by Sony α6000 with identical light conditions: **(a)** ISO 100, **(b)** ISO 25600.

In digital consumer-grade cameras, the noise level is affected by adjusting the ISO parameter. The term originates from the sensitivity of analog films – the lower the ISO value, the lower the film's sensitivity. This term is still used in digital cameras, however, the value no longer represents the coarseness of the film grain but rather the digital signal amplification level. Digital cameras usually offer the ISO ranges from 100 to 25600 or sometimes even higher. By increasing the ISO level the amount of light needed for adequate exposure is reduced, but the amount of image noise is increased. Figure 2.1 demonstrates the deteriorating effects of high levels of ISO on a digital image.

It is apparent that digital images are corrupted by various different noise sources that are hardly deterministic. The increased demand for small portable devices capable of digital photography, such as smartphones and compact digital cameras, sees a surge in the need for noise reduction techniques as those devices suffer greatly from these hardware-induced causes of digital image noise.

2.2 Traditional noise reduction approaches

Image denoising techniques have attracted much attention in the recent 50 years due to the surging popularity of digital imagery. However, the task of reducing image noise remains a challenging and open task. The main reason is that from a purely mathematical perspective, image denoising is an inverse problem with a non-unique solution. The major challenges of noise reduction in digital images are as follows:

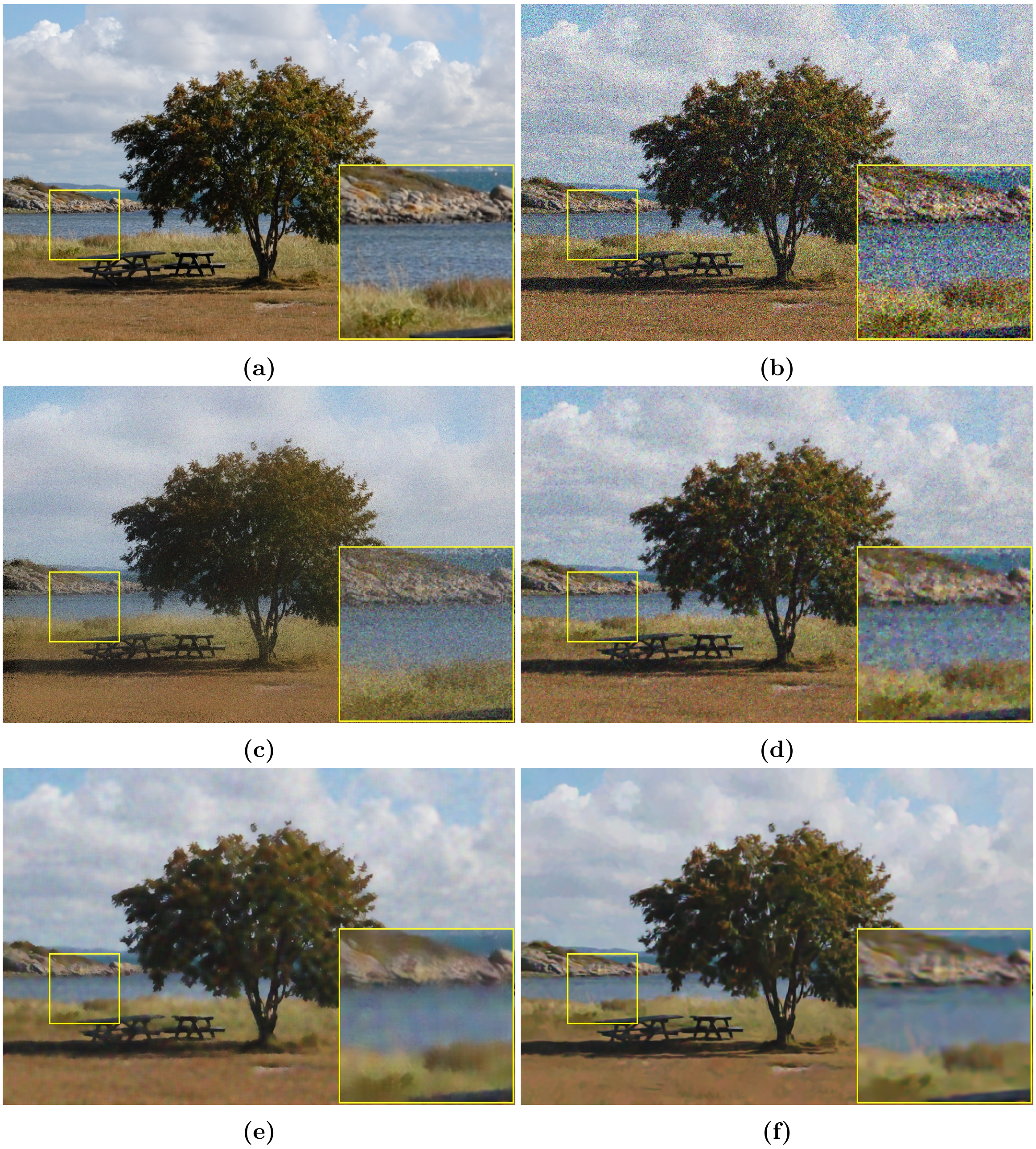
- Edges in the image should be preserved with minimal blurring
- Flat areas in images should be smooth and uniform
- Textures should be retained, especially high-frequency ones
- New artifacts should not be generated

Classical denoising methods work in the spatial domain, meaning that the aim of these methods is to remove noise by calculating the values of each pixel based on the correlation between pixels/patches in the original image. Since the use of filters is prevalent in image processing, a substantial number of image filters have been used for image denoising [14]. Spatial filters can be further divided into two types: linear filters and non-linear filters. The originally used linear filters achieved reasonable success in reducing image noise, however, approaches such as mean filtering or Weiner filtering [4] suffered from image over-smoothing, edge blurring, and poor image texture preservation. Non-linear filtering approaches, such as Bilateral filtering [41] show improvement over the linear filters, however, this comes at a cost of efficiency. Spatial filters utilize low pass filtering, which follows the presumption that the image noise occupies a higher region of the frequency spectrum. This means that while the noise reduction of spatial filters is reasonable, the loss of high-frequency details and image blurring is inevitable.

One of the most influential noise reduction methods is total variation (TV) regularization [36]. The main principle of this approach is based on a presumption that noisy images have high total variation, which means that the integral of the absolute image gradient is high. According to this principle, reducing the total variation of the image should remove unwanted noise whilst preserving important details such as edges. It has had tremendous success in the field of denoising, particularly because of its ability to effectively calculate the optimal solution while preserving sharp edges. However, even this method suffered from some major drawbacks, such as over-smoothing and loss of contrast. To improve the performance of the TV-based regularization model, a tremendous amount of research has been conducted on image smoothing by using partial differential equations [32].

While local denoising methods, which take into account only a group of pixels surrounding a target pixel, have low time complexities, their performance on higher levels of image noise is limited. The reason for this is that the correlation between neighboring pixels is severely disturbed in high noise conditions. Pioneering work on non-local means (NLM) [6] strived to solve this shortcoming by using weighted filtering across all pixels in the target image. Thanks to this approach the NLM can make full use of the information provided by the given images, which results in much greater post-filtering image clarity and lower loss of detail than its local denoising counterparts. Since its conception, many studies on improving this approach have been conducted focused on speed and denoising performance improvements [28][13].

Apart from methods working in the spatial domain, methods working in the so-called transform domain have seen a lot of success in recent years. In contrast with spatial domain filtering, the transform domain methods first transform the image data to another domain and then apply a denoising procedure on the transformed image data according to the different characteristics of the image and image noise. One of the most influential methods utilizing this approach is the block-matching and 3D filtering (BM3D) method [11]. This non-local method uses the strategy of grouping image patches with similar characteristics into three-dimensional groups (block matching) which are later transformed, filtered (usually by Weiner filtering), and reconstructed to form the output image. This method



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 2.2: Image denoised by several different algorithmic methods: **(a)** Original image, **(b)** Noisy image with Gaussian noise of $\sigma = 50$, **(c)** Bilateral filtering applied, **(d)** TV filtering applied, **(e)** NLM filtering aplied, **(f)** BM3D applied.

provides excellent results and is often used as a benchmark algorithm for new denoising methods due to its great performance.

Figure 2.2 demonstrates image denoising results of selected methods discussed above to highlight their denoising capabilities and shortcomings. It needs to be noted that due to the need for manual tuning of parameters for each method the shown results may not be fully optimal and serve for illustrative purposes only.

2.3 Commonly used evaluation metrics

As stated earlier, digital images are subject to a wide variety of distortions during the capturing process. In order to quantify the visual image quality, several techniques are used which will be described in this section.

For applications in which images are ultimately to be viewed only by humans, such is usually the case in digital photography, the only “proper” way of quantifying the image quality is through subjective evaluation on human test subjects. Subjective methods operate without reference, meaning that the ground truth image is not needed for evaluation. In practice, however, subjective evaluation is inconvenient due to being time-consuming, expensive, and labor-intensive.

On the contrary, objective evaluation is based on comparisons using explicit numerical criteria. Several references are possible such as the ground truth image or the use of prior knowledge, which makes it more practical than subjective evaluation. Objective image quality metrics can be divided by the availability of an original (noise-free) image. Most widely used approaches are known as *full-reference*, meaning that the original image needs to be provided in order to compute the visual quality. However, in many situations, the reference image is simply not available and in turn, a *no-reference* quality assessment approach is needed.

MSE and PSNR

The most widely used full-reference quality metric is the mean squared error (MSE), which is computed by averaging the squared intensity differences between the reference and noisy image pixels:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{2.1}$$

where m and n represent the image dimensions, I represents the noise-free image, and K its noisy counterpart. Along with MSE, a derived metric called peak signal-to-noise ratio (PSNR) is widely used in most current publications. This metric represents the ratio between the maximum possible power of a signal and the power of corrupting noise. It is calculated in the following way:

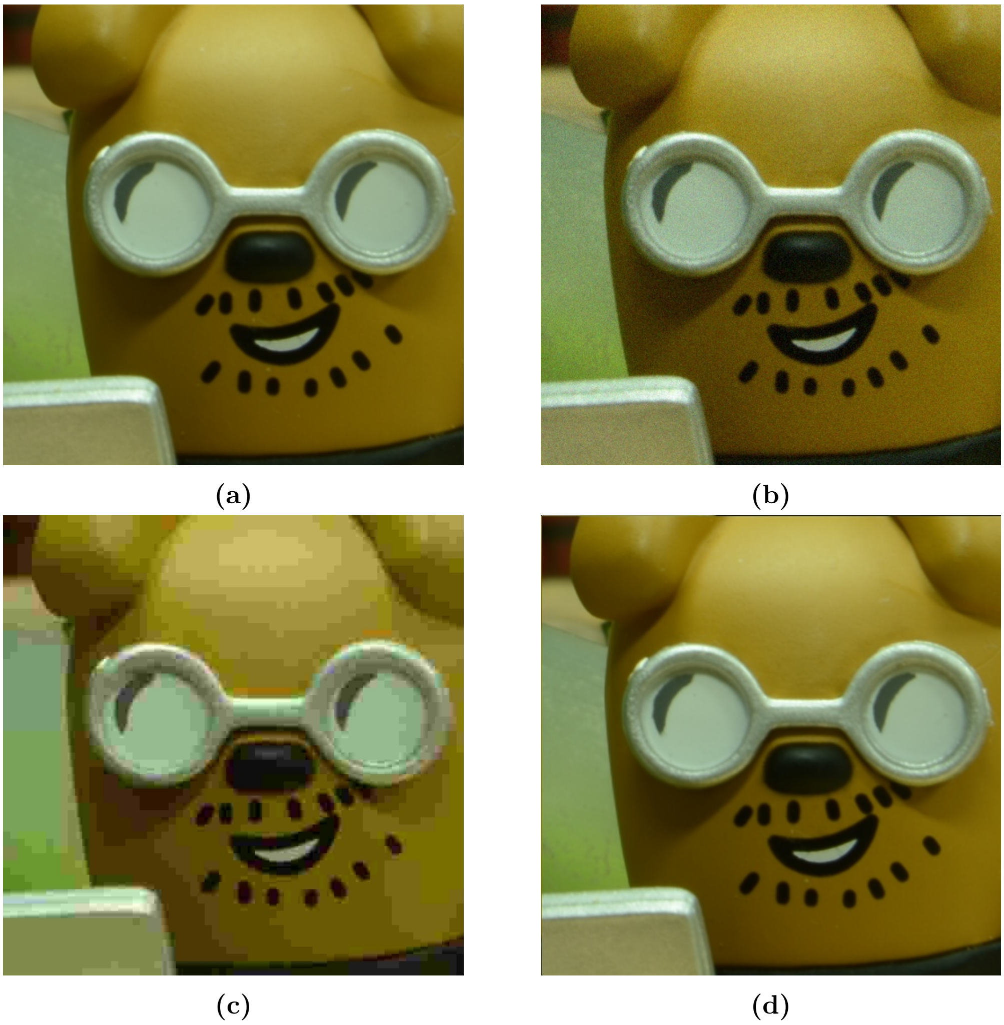
$$PSNR = 10 \times \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{2.2}$$

where MAX represents the maximum possible pixel value of the image and MSE represents the mean squared error. Both Equations 2.1 and 2.2 account only for monochrome images for the sake of simplicity. Both metrics are inverse to each other in their representation of image quality – when image quality decreases, MSE rises and PSNR approaches zero, and vice versa.

Both metrics are appealing because of the fact that they are simple to calculate, have clear physical meanings, and are mathematically convenient for optimization methods. However, it is known that these metrics do not correlate well with perceived visual quality.

SSIM

Human perception of image quality is more closely tied to factors such as structural similarity which metrics such as MSE or PSNR do not take into account. Alternative metrics, such as the structured similarity index metric (SSIM) [43], address these issues by taking into accord the degradation of structural information which should more closely correlate with the perceived visual quality. The SSIM metric is considered greatly influential due to being one of the highest cited papers in the image processing and video engineering fields. However, current publications often use PSNR as the only evaluation metric and neglect other alternatives that could be more suitable. Figure 2.3 demonstrates the differences between PSNR and SSIM metrics on various image distortion types – all images share similar values of PSNR but different values of SSIM.


(a)

(b)

(c)

(d)

Figure 2.3: Image from the CC dataset with various distortions applied to demonstrate differing values of PSNR and SSIM. Values are presented in the format [PSNR/SSIM]: **(a)** Original image [∞/1], **(b)** Gaussian noise of $\sigma = 10$ applied [28.48/0.49], **(c)** JPEG compression with quality factor of 7 applied [29.07/0.79], **(d)** Shift by 1 pixel in the x and y axis applied [30.62/0.96].

No-reference metrics

In contrast, the no-reference metrics make use of prior knowledge about anticipated distortions in the form of training examples and corresponding human opinion sources to be able to determine the image quality. Some of the widely used metrics are NRQM [26], NIQE [31], and PIQUE [33]. These metrics are especially useful when the ground truth images are not available.

Chapter 3

Overview of the current state

This chapter outlines the currently used state-of-the-art denoising methods based on deep neural networks. Along with their brief introductions, commonly used datasets that are used for the purposes of training and testing these methods are also presented.

3.1 Deep neural networks for image noise reduction

Deep neural networks have seen a big upsurge in popularity for image restoration tasks in recent years. This rise in popularity is mainly caused by the easily accessible large-scale datasets, efficient training implementation on modern powerful GPUs, and various advances in deep learning methods [45].

The state-of-the-art deep learning denoising methods are typically based on convolutional neural networks (CNN). The use of a CNN for image denoising can be traced back to [23], where a five-layer network with sigmoid nonlinearity was proposed, which showed improvements over the then-state-of-the-art denoising methods. However, CNN-based methods still faced problems with high memory requirements and computational complexity. To address these problems, deep network architectures with small filters were preferred to improve the performance and reduce computational costs. One of the notable networks, the VGG [37] stacked multiple convolution layers with small kernel sizes to a great effect. With this success, research has turned to deeper networks in hopes of greater denoising performance. Although deeper networks are effective for image applications, they suffer from two major drawbacks: first, if the depth of the network is high, problems with vanishing or exploding gradients may occur; second, problems such as overfitting may affect networks with very wide architectures. To overcome these issues, the ResNet [19] utilized residual learning operation which helped to alleviate some of the shortcomings. Very influential work by Zhang *et al.* [47] introduced techniques of residual learning and batch normalization [22] incorporated into a feed-forward denoising CNN-based network, called the DnCNN, for the first time. Specifically, it utilized the fact that residual learning and batch normalization benefit from each other to boost denoising performance.

Some networks such as BRDNet [39] or DudeNet [40] utilized two sub-networks to enable greater feature extraction while utilizing dilated convolutions to decrease performance penalties of wider networks. Moreover, BRDNet experimented with batch renormalization [21] which allows for the use of smaller mini-batches unlike BN, which is invalid for smaller batch sizes. Both methods also presented promising results on real image denoising in their publications.

Due to the lack of suitable training data for real image processing, generative adversarial networks (GAN) were developed in order to address this issue. In general, GAN consists of a discriminative network and a generative network. The discriminative network is trained on determining if an image sample is real or generated while the generative network is trained to produce samples good enough to fool the discriminative network. The resulting distribution is aimed to be as close as possible to the distribution for real data. While it is well known that training GAN is tricky and unstable, works by Chen *et al.* [7] and Guo *et al.* [18] showed promising results on synthetic and real image denoising.

Loss functions

One of the most influential factors of deep neural networks is the loss function used in the training process. Despite this fact, the choice of this function is usually not discussed in most recent works. Mean squared error (MSE), an *L2*-norm measure, is arguably one of the most dominant error measures. Some of the main reasons for its popularity are the fact that it is convex, computationally effective, and differentiable, which are very convenient properties for optimization problems. However, as discussed in Section 2.3, the MSE does not always correlate well with the perceived visual quality due to working on a per-pixel basis.

Metrics such as SSIM are perceptually motivated, which should more closely correlate with the desirable perceived quality. Works such as Couturier *et al.* [10] employed a combination of *L1*-norm and SSIM as a loss function, which delivered impressive results mainly in higher levels of noise. Work by Johnson *et al.* [24] demonstrated the benefits of using perceptual loss instead of traditional per-pixel loss on several existing methods which showed improvements both in speed and visual quality.

3.2 Commonly used datasets

It is no secret that datasets are very important factors dictating the performance of image denoising methods based on deep neural networks. The proper choice of training dataset can greatly affect the denoising performance and thus this task should not be underestimated. The datasets used for training and testing can be divided into two categories: datasets for synthetic noise removal and datasets for real image denoising.

Datasets for synthetic noise removal, such as Additive White Gaussian Noise, are generally more abundant due to the fact that no ground truth image needs to be generated – to generate clean/noisy pairs, the clean image is augmented by artificially adding the image noise. This means that virtually any image dataset can be used for this task which provides great flexibility. Some of the most used training datasets are the BSD300 [29], Waterloo Exploration Database [27], and Urban100 [20]. For testing, the datasets used are generally of smaller size and are never present in the training phase. Some of the most frequently used testing datasets are the BSD68 [30], Kodak24 [15], McMaster [48], or Set12 [47]. In some cases, grayscale versions of the datasets are used to train and test models focused on monochromatic image denoising.

Datasets containing noisy photographs are on the other hand way less common. This is due to the fact that for the purposes of training a deep neural network, a ground truth image has to be provided in order to compute the loss function by a full-reference metric. For real images the process of obtaining a clean/noisy image pair is complicated. Some earlier approaches tried to solve this problem by capturing images twice, once with a high

ISO level that caused higher noise and once with a low ISO level to obtain a clean image with low levels of noise [2][35]. This approach has several problems: the images tend to have different characteristics due to the fact that the camera needs to compensate for the ISO levels in either change of aperture or shutter speed which can change the characteristics of the image and the scene can change over time due to movement or changing light conditions (especially if the light source is artificial and is powered by alternating current). These problems can be partially solved by capturing large amounts of noisy images and generating the ground truth image by image averaging and subsequent color correction, dead pixel correction, and other forms of post-processing [34][44][1]. These techniques produce great results that are being gradually adopted for the training and testing of real image denoising methods. However, due to the relatively complicated process that is needed for the generation of a real image dataset the availability of public datasets is still limited. This lack of variety results in incomplete training data that is limited to only a selection of camera types, noise levels, and scene settings which can reflect on the trained method's performance under different scenarios.

Chapter 4

Proposed solution for noise reduction in photos

This chapter introduces the proposed solution for noise reduction in photos. First, a task definition is presented that outlines the goals of this part of the thesis. Afterward, selected noise reduction methods that will be used are presented. Finally, the used datasets for training and testing said methods are presented and demonstrated.

4.1 Task definition

The first objective is to implement two state-of-the-art denoising methods based on deep convolutional neural networks and evaluate their relative image denoising performance on several public datasets. Results can show how big of an impact the different method architectures and method novelty have on denoising performance, run times, and detail preservation. The results can be also used to independently validate the claims made by the methods' authors about their noise reduction performances and provide additional data on originally unexplored test cases, such as denoising of real photos.

The second objective is to determine the importance of datasets containing realistic image noise in the task of real image denoising. Existing noise reduction methods mostly rely on simple noise assumptions, such as Additive White Gaussian Noise (AWGN), and ignore the task of noise removal from photographs. As described in Chapter 2.1, the degradation of real images is affected by several different factors and noise types which poses a question if models trained on AWGN can produce competitive results in real image denoising. Due to the fact that a substantial amount of current methods provide results solely on AWGN reduction, it is not easy to find out the answer just from publications alone. The results from testing can provide insight into the importance of using real noisy image datasets or simulating realistic image noise in boosting the noise reduction performance in real photos.

4.2 Selected methods of study

The goal was to select methods that offer state-of-the-art denoising performance and utilize different architectures and network complexity. Emphasis was also put on selecting methods that differ in novelty to study the advancements made in the field of noise reduction in digital images. For this purpose, two different methods were selected that meet the desired criteria – the DnCNN [47] and BRDNet [39].

DnCNN

The first studied method is a state-of-the-art denoising convolutional neural network proposed by Zhang *et al.* [47] called DnCNN. Since its publication in 2017 this paper was cited over 2690 times and is regarded as a highly influential work in its field. Due to its great performance in noise reduction, it is still widely used as a benchmark for newly proposed denoising methods.

This network utilizes the residual learning technique to predict the image noise that is corrupting a noisy image. In contrast, discriminative denoising methods aim to directly predict the original image instead.

$$x = y - \mathcal{R}(y) \tag{4.1}$$

The equation 4.1 shows the process of residual learning where x is the underlying deterministic noise-free image, y is the noise-corrupted image and $\mathcal{R}(y)$ is the residual mapping of the noise-corrupted image to obtain the underlying image noise.

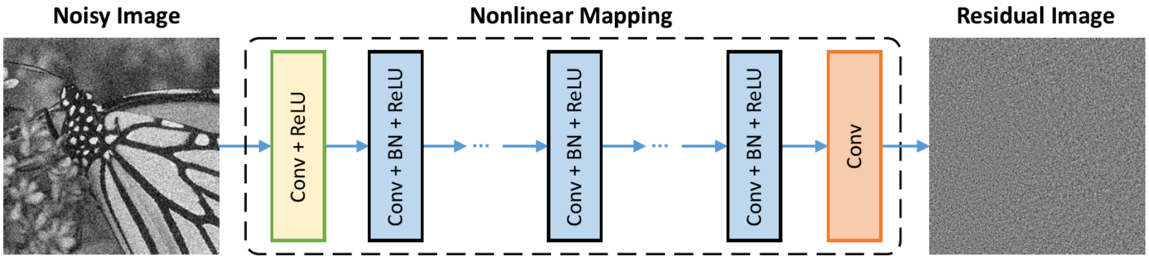


Figure 4.1: The architecture of the DnCNN network. The image was taken from the original publication [47].

The architecture of this method can be seen in Figure 4.1. Specifically, the architecture consists of three types of layers:

1. **Conv + ReLU**: The first layer utilizes 64 filters of size $3 \times 3 \times c$ that are used to generate 64 feature maps, and rectified linear units (ReLU) for nonlinearity. Here c represents the number of image channels, i.e. $c = 3$ for color images and $c = 1$ for grayscale images.

2. **Conv + BN + ReLU**: Layers $2 \sim (D-1)$ each contain 64 filters of size $3 \times 3 \times 64$ with batch normalization [22] layer added between Conv and ReLU layers.

3. **Conv**: For the last layer, c filters of size $3 \times 3 \times 64$ are used to reconstruct the output.

The number of inner layers is dependent on the use case to provide a trade-off between complexity and performance – for image denoising on a known noise level inner depth of $D = 15$ is selected while for blind image denoising depth of $D = 18$ is selected. For the reduction of boundary artifacts, the authors propose using zero padding. In summary, the DnCNN model has two main features: the residual learning technique is adopted to learn $\mathcal{R}(y)$, and the batch normalization layer is used to increase noise reduction performance and training speed. In particular, both residual learning and batch normalization benefit from each other and in turn provide even faster training speeds and higher denoising performance.

The authors presented test results from denoising of both grayscale and color images in the task of AWGN denoising. The tests included models trained on both fixed noise levels and unknown noise levels where DnCNN showed state-of-the-art results and consistently outperformed traditional algorithmic methods such as BM3D [11] and WNNM [17]. Experiments on real noisy images were briefly mentioned but a thorough evaluation of real image denoising was missing. The authors also failed to address training on real noisy images, so this area of interest remained untouched in the original publication.

In the paper, the authors also presented the ability to train a single DnCNN model for three image restoration tasks – image denoising, super-resolution, and JPEG deblocking. This single DnCNN model showed impressive and competitive results in all three tasks. However, since the focus of this thesis is solely on image denoising, only models dedicated to noise reduction exclusively were experimented with.

BRDNet

For the second method, a batch-renormalization denoising network (BRDNet) from Tian *et al.* [39] was chosen. This novel network proposed in 2020 was cited 159 times at the time of writing this thesis and promises performance improvements in denoising capabilities over other state-of-the-art image denoising methods.

This network utilizes a unique architecture that concatenates two networks in order to increase the total width of the network and in turn increase denoising performance. Figure 4.2 shows the BRDNet's architecture outline. The top network consists of 17 layers

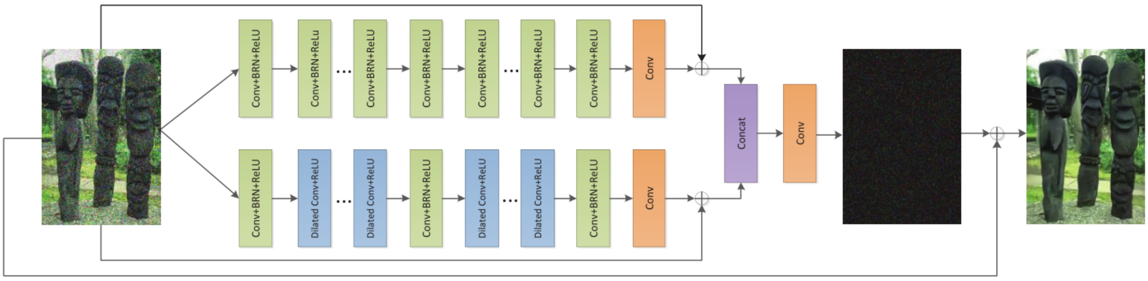


Figure 4.2: The architecture of the BRDNet network. The image was taken from the original publication [39].

that include Convolution layers, Batch Renormalization Layers (BRN) [21], and ReLU. The layers $1 - 16$ consist of Conv + BRN + ReLU and the last layer is Conv only.

The bottom network includes similarly to the top one 17 layers. The first, ninth, and sixteenth layers are Conv + BRN + ReLU. Layers $2 - 8$ and $10 - 15$ contain Dilated convolution and ReLU. The last layer is Conv to reconstruct the output.

The Batch Renormalization Layer is used to address small-batch and internal covariate-shift problems. The ability to use smaller mini-batches is especially beneficial for low-configuration hardware due to this method being computationally demanding. Additionally, it utilizes dilated convolutions that allow for a larger receptive field size while saving computational costs.

The authors presented results from color and grayscale AWGN denoising on known fixed image noise. BRDNet was shown as a superior method to DnCNN and other competing methods due to its ability to better preserve image details and minimize the introduction of artifacts. Additionally, the authors decided to include results from BRDNet trained on real noisy images where it showed great potential and produced results that outperformed DnCNN by a margin of almost 3 dB of PSNR.

It needs to be noted that an even newer method by the same authors called DudeNet [40] was considered for use in this thesis. However, upon further inspection some strange nuances were discovered – the DudeNet publication did not reference its predecessor BRDNet in any part of the paper nor did it include it in any of the performance comparison tables. What's more, the older BRDNet achieved slightly *better* denoising results than the newly proposed DudeNet. It appears that perhaps the authors wanted to present their new method in a favorable light and in turn, did not mention the slightly older and better-performing BRDNet. Due to these strange circumstances, the more established BRDNet was chosen instead.

4.3 Used datasets

This section summarises and describes datasets used for training and testing the studied noise reduction methods. The training and testing datasets were selected to match the datasets used by the authors to provide an accurate comparison.

Training datasets

For training models on color AWGN denoising, the Berkley Segmentation Dataset (BSD300) [29] was selected. The BSD300's training set contains 200 color images with the sizes of 428 × 381 pixels (or 381 × 428 for portrait images) in png format. This dataset was originally created for the purposes of image segmentation tasks, however, it has been widely adopted for the training of image restoration models. The dataset is composed of a large variety of images ranging from natural images to object-specific images such as plants, people, and food. Some of the images can be seen in Figure 4.3.

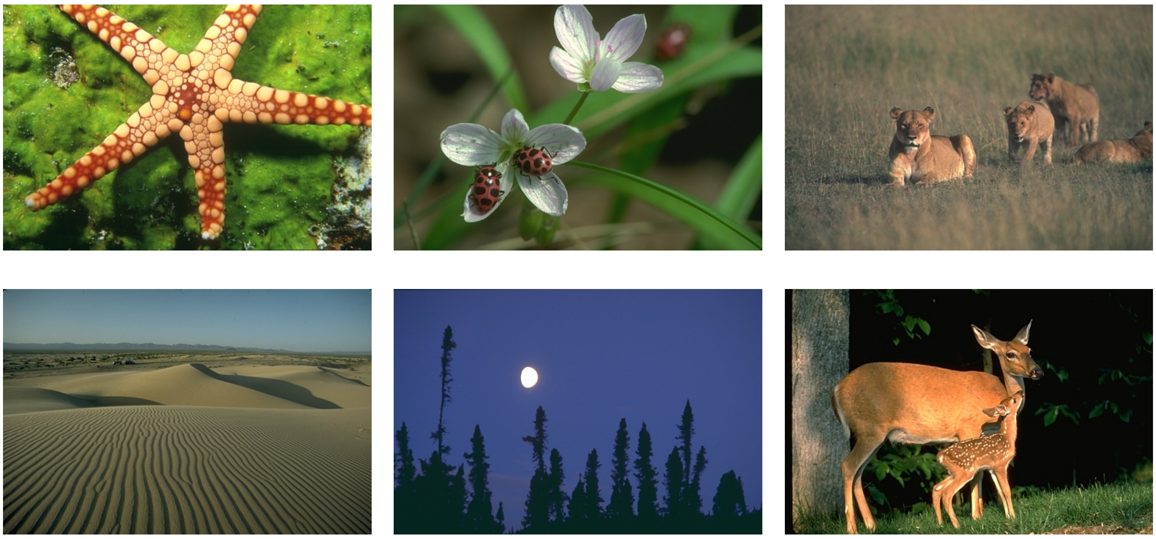


Figure 4.3: 6 natural images from the BSD300 dataset.

The training of the real image noise reduction models was performed on the PolyU [44] dataset. It contains 100 real noisy images captured by different cameras (Canon 5D Mark II, Canon 80D, Canon 600D, Nikon D800, and Sony A7 II) with various image sensor sizes, ISO values, and scene settings. The dataset contains image pairs of noisy and ground truth images that were generated by image averaging. Each image has a resolution of 2784 × 1856 pixels and comes in the `jpg` format. This dataset was chosen due to being featured in the BRDNet publication and due to its higher quality than its counterparts, such as RENOIR [2], which obtain ground truth images by capturing low ISO images.

Testing datasets

For the testing of AWGN reduction in color images the datasets CBSD68 [30] and Kodak24 [15] were used. The CBSD68 dataset contains a selection of 68 color images from the Berkeley segmentation dataset with sizes of 428 × 381 pixels. The Kodak24 dataset contains 24 color images with sizes of 768 × 512 pixels. Both selected datasets are widely used for measuring the denoising performance of deep neural networks.

The real image noise reduction models were tested on the CC [34] dataset, which contains 15 real noisy images captured by three different consumer-grade cameras (Nikon D800, Nikon D600, and Canon 5D Mark III) with different ISO values (1600, 3200, and 6400). This dataset contains pairs of noisy and ground truth images that were obtained similarly to the PolyU dataset by averaging multiple scene instances. All 15 images can be seen in Figure 4.4.

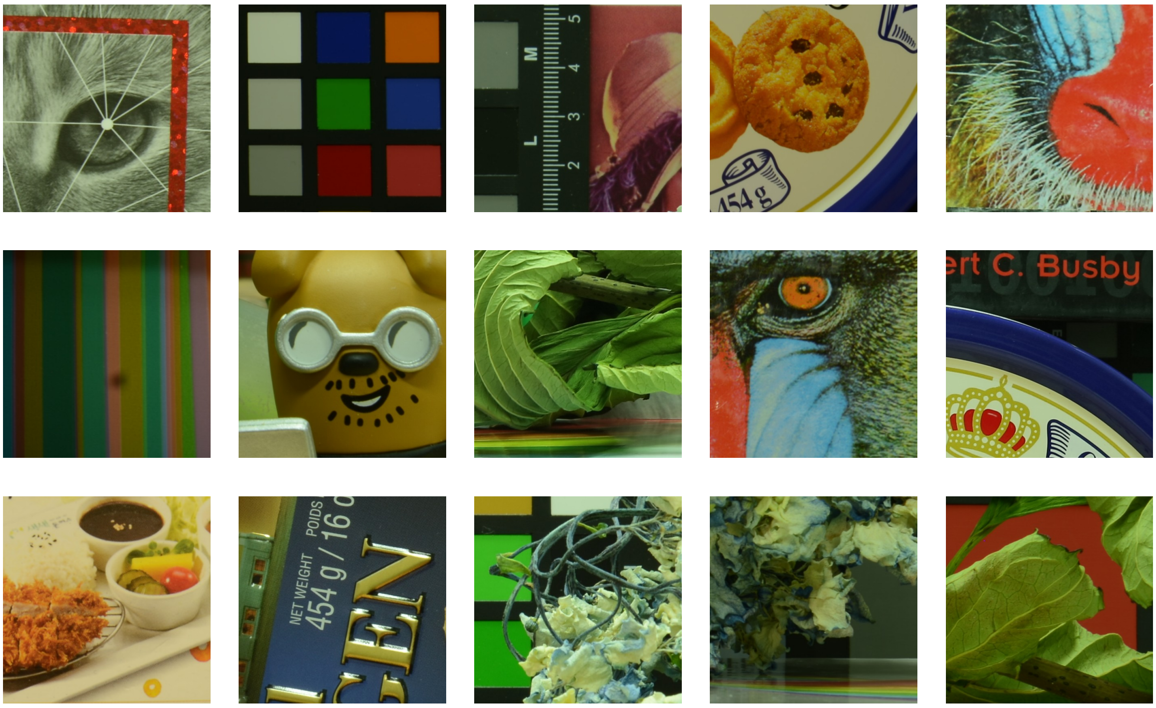


Figure 4.4: 15 images from the CC dataset.

Chapter 5

User testing design

This chapter introduces the proposed user testing experiment for image quality evaluation by humans. Firstly, a task definition is presented along with an introduction to the dataset constructed specifically for this task. Later, the design of the tool used by the test subjects to rank the images is presented and discussed.

5.1 Task definition

All noise reduction methods aim to achieve a common goal – improvement in the quality of the original image corrupted by noise. For the purpose of evaluating the improvement, metrics such as PSNR and SSIM (described in detail in Section 2.3) are used to provide numerical values that indicate how much the output image correlates with the original image. These metrics are widely used in the field and provide a solid basis for comparing the performance of different noise reduction methods.

However, it is known that the concept of overall image quality is rarely unidimensional and different attributes such as blur, noise, and artifacts all influence the perceived image quality [42]. Motivated by this fact an experiment was set up with the aim to rank the noise reduction performance of different methods in different conditions based on human visual perception.

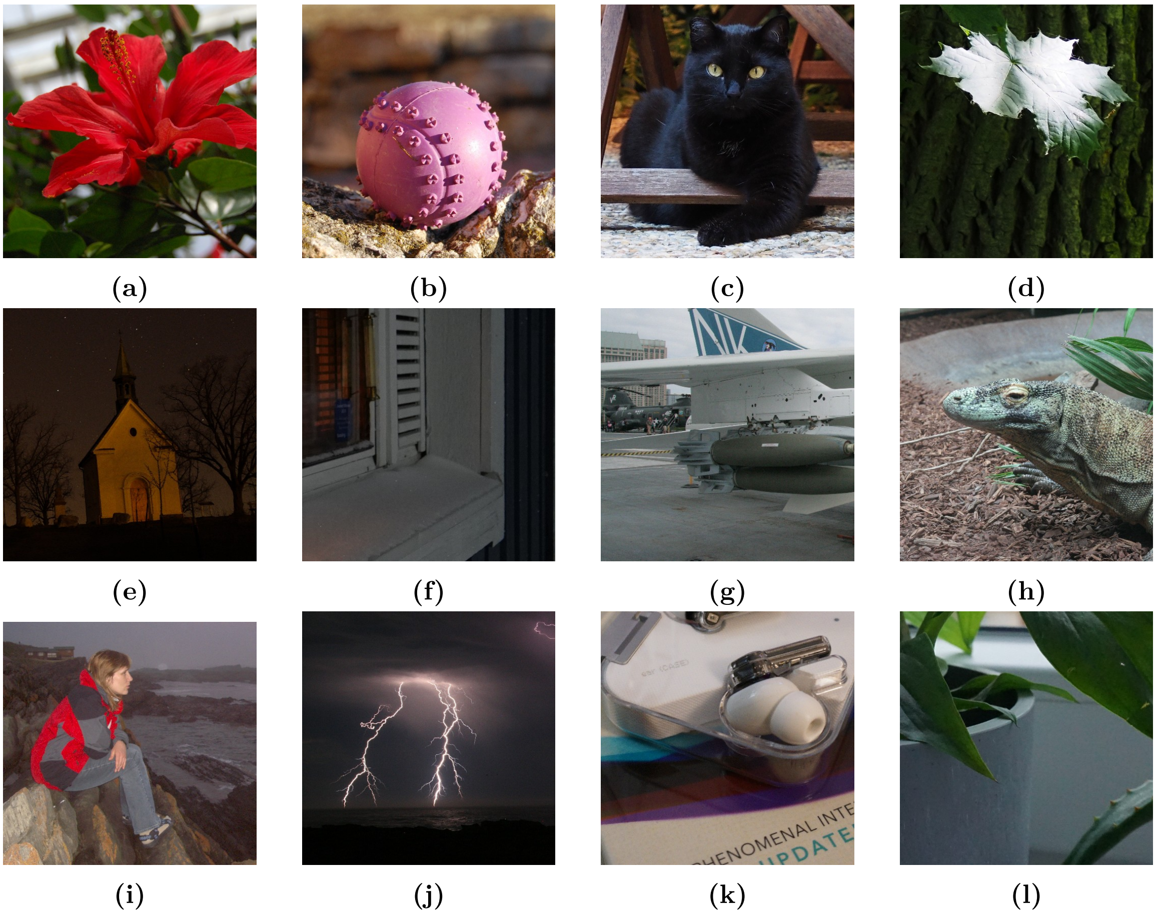
5.2 Used dataset

In order to evaluate the perceived visual quality of different noise reduction methods, a suitable dataset needs to be selected. Publicly available datasets containing real noisy images such as PolyU [44], CC [34], or SIDD [1] were considered, however, all were ultimately rejected due to small variability in scene settings or too low variability in used camera types. For these reasons a custom dataset, further referenced only as the Survey Dataset, was constructed to meet the needs of this test.

It is known that the main factors that affect the image corruption level by digital noise are lighting conditions, ISO level, temperature, image sensor size, and sensor manufacturer. However, the characteristics of image noise can differ in relation to different light conditions [16]. For this reason, the dataset should provide adequate variability in image light levels to account for this phenomenon.

Another aspect that was considered while constructing this dataset was the scene setting. Homogenous environments, such as in PolyU where all images are indoor images of similar nature, do not have to provide objective grounds for comparison due to the lack of variability. It is nearly impossible to construct a dataset that covers all possible scene types of digital photography, but the aim was to provide as much variability in scene types as possible to facilitate fair and objective comparison.

In total, 12 images were selected for the Survey Dataset (Figure 5.1). The total number of images needed to be low in order to keep the attention of the test subject at a maximum – 12 was deemed to be acceptable in this aspect. Half of the images in the dataset are considered as low-light (i.e. with either artificial light or low intensity of natural light) and the other half are deemed as high-light for opposite reasons. The ISO levels are in the range of $[200, 3200]$. The scene types of selected images contained nature, architecture, animals, people, sceneries, and man-made objects. The dataset features images from 6 different consumer-grade cameras. Each camera was represented by 2 images in the dataset. The camera sensor formats, resolutions, and years of release can be found in Table 5.1. Each image was cropped to 1500×1500 pixels to normalize the size for easier evaluation. All images were kept in their original file format `jpeg` and sRGB color space.



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

(i) **(j)** **(k)** **(l)**

Figure 5.1: 12 images from the Survey Dataset captured by various cameras: **(a)(b)** Nikon D40, **(c)(d)** Olympus E-M5 Mark III, **(e)(f)** Sony α7, **(g)(h)** Canon EOS 350D, **(i)(j)** Pentax Optio S4, **(k)(l)** Sony α6000.

A high emphasis was also put on selecting images from various camera manufacturers. In total, 5 different camera manufacturers were represented in this dataset. The reason for this is the fact that camera chips have different image noise characteristics depending on the manufacturer which may affect the denoising performance of studied methods [16]. The final results from the Public Survey, which was conducted using this dataset, can be found in Section 7.4.

Camera name	Sensor format	Sensor resolution	Year of release
Canon EOS 350D	APS-C	8 Mpix	2005
Pentax Optio S4	1/2.5	4.23 Mpix	2003
Sony α6000	APS-C	24.3 Mpix	2014
Nikon D40	Nikon DX	6.1 Mpix	2007
Olympus E-M5 Mark III	4/3	20 Mpix	2019
Sony α7	Full-frame	24 Mpix	2014

Table 5.1: The digital cameras used in the Survey Dataset and their respective parameters.

5.3 Survey tool

The gathering of data from the general public is a common task and a lot of tools exist to automate this process via online forms and questionnaires. However, for the task of comparing and evaluating the noise reduction performance of different methods these solutions were deemed as insufficient due to their poor ability to facilitate image comparisons. Because of this, a custom surveying tool with proper image comparison capabilities was designed to help with automating the data gathering process.

The tool's purpose is to present the user with 12 image samples, gather the user's responses and store them for later evaluation. Each image sample consists of three individual images – the original image, the image produced by DnCNN, and the image produced by BRDNet. In order to prevent selection bias, each image sample has the order of the three images randomized. The user is required to rank the three images in each sample from best to worst in terms of perceived visual quality.

The Survey tool was implemented as a web application. This choice was made due to the fact that the web is a platform that is available to the vast majority of people which is desirable when conducting this type of experiment. The user interface was in turn designed to meet the needs of the web browser. The main goal was to provide as much screen space to the images as possible to minimize the amount of distracting factors for the user. Additionally, a "magnifying glass" element was designed that enhances a user-selected area of interest in all three images simultaneously to help spot subtle differences in the images by the user. Details about the final implementation of this tool are provided in Section 6.3.

Chapter 6

Implementation

This chapter describes the technologies that were used throughout this thesis. Implementation details of both noise reduction methods and the public survey tool are also presented later in this chapter.

6.1 Used technologies

The Integrated Development Environment (IDE) of choice for implementation of all parts of this thesis was Visual Studio Code[1]. This choice was made due to its broad support of countless programming languages, simplicity, platform independence, and performance. Due to its popularity, a wide range of extensions can be installed that offer support for more language features, debugging, and other features that help with the development process.

Two main programming languages were utilized in this thesis: Python for implementation, training, testing, and evaluation of the two denoising methods and TypeScript for implementing the Public Survey Tool.

When implementing a new piece of software, the ability to rapidly prototype smaller pieces of code is very useful and helps with early error prevention. For this reason, the Jupyter computational environment was used to rapidly prototype Python code directly in Visual Studio Code thanks to the Jupyter extension[2].

Any project of moderate size should utilize some form of versioning system to keep track of changes, manage version control and have access to the history of the project. For these reasons, a popular versioning system Git was chosen. This choice was made because of its clear command line syntax, wide adoption in the field, and previous experience with using this tool in school and work tasks. GitHub was chosen as a Git hosting service, which enables users to store their projects versioned under the Git system remotely for free. This prevents accidental data loss and enables simultaneous work on multiple devices while keeping up-to-date.

An honorable mention also goes to GitHub Copilot[3], a tool developed by OpenAI in collaboration with GitHub intended to help programmers with writing code which was used in form of a Visual Studio Code extension[4]. The following paragraph describing what GitHub Copilot is was taken from the official website:

[1] https://code.visualstudio.com/

[2] https://marketplace.visualstudio.com/items?itemName=ms-toolsai.jupyter

[3] https://copilot.github.com

[4] https://marketplace.visualstudio.com/items?itemName=Metatype.copilot-vscode

"*GitHub Copilot is an AI pair programmer that helps you write code faster and with less work. GitHub Copilot draws context from comments and code and suggests individual lines and whole functions instantly.*"

Copilot is powered by OpenAI Codex [8], an artificial intelligence model used to process natural language, which is based on GPT-3 [5], a widely known and popular autoregressive transformer model for natural language processing also developed by OpenAI. Copilots model was trained on a selection of English language as well as source code from public GitHub repositories, which enables it to provide assistance in a wide range of programming languages. This tool was provided to me in a closed technical preview version, which was granted to a limited number of signed-up users for testing purposes. During the implementation phase of this thesis, Copilot was tested on the Python, TypeScript, HTML, and CSS languages. Its performance during testing was remarkably impressive. The ability to generate syntactically correct code in a language the user is less familiar with, accurately predict and suggest the next line the user wants to write, or help with generating boilerplate code from scratch saves a lot of time and helps the end-user to focus on the problem and less so on the implementation. It needs to be noted that the technology is still evolving and the overall accuracy of the model is limited and the user's supervision is necessary. The overall experience with this tool was very positive and I believe that this technology has a great potential.

6.2 Noise reduction methods

The programming language of choice for implementing both noise reduction methods was Python. The main reason for this choice was its syntactical simplicity with minimal boilerplate code required, which greatly reduces the time needed for implementation. Moreover, the Python language offers powerful libraries suited for machine learning tasks. The Pytorch library was chosen for the implementation of both studied noise reduction methods. Pytorch is a free and open-source deep learning framework that is both flexible and powerful. Thanks to its built-in CUDA[5] support, the Pytorch framework offers seamless GPU acceleration out-of-the-box. Two other Python libraries that were heavily utilized were NumPy for image manipulation and evaluation and Matplotlib for output visualization.

The first studied method, the DnCNN, was implemented in the Pytorch framework in accordance with the original paper [47] and the official implementation in a GitHub "monorepo" [6] created by the paper's authors. The implementation utilized the basic building blocks offered by the Pytorch framework. The total depth of the network was set to 20 as per the authors' recommendation for blind Gaussian noise reduction.

The BRDNet's implementation was based on the method's paper [39] and the official implementation[7]. However, the implementation was written in the Tensorflow framework and thus had to be translated to the Pytorch environment. A third-party library[8] was used to include the Batch Renormalization [21] layer in the method's architecture as proposed by the authors.

The Pytorch framework also allows for custom data loader implementation. Data loaders are Pytorch primitives that facilitate iteration over a dataset. Several optimizations, such as preloading all dataset entries to minimize the costly IO operations, were made to

[5] https://developer.nvidia.com/cuda-toolkit

[6] https://github.com/cszn/KAIR

[7] https://github.com/hellloxiaotian/BRDNet

[8] https://github.com/ludvb/batchrenorm

decrease the training and testing times. Even marginal improvements in speed can save a substantial amount of time in tasks such as training that often take tens of hours to complete. Moreover, custom implementation was also needed to perform data augmentation and in certain cases the addition of artificial noise. Data augmentation was performed on each sample and consisted of randomly cropping the image to the desired dimensions and subsequently flipping and/or rotating it to prevent model overfitting.

6.3 Public survey tool

The public survey tool was implemented as a web application due to the fact that nowadays the web is the most popular and accessible platform for the general public. The front-end part of the app was implemented in the Angular[9] framework. This TypeScript-based framework is one of the most popular front-end frameworks used today due to its robustness, scalability, and maturity. It is usually used for larger-scale web applications and thus this choice may appear to be unnecessarily robust for this task. However, this framework was chosen due to previous experience with it from other projects that I was involved in. Due to the relatively simple structure of this web application, the back-end portion of the app did not have to perform any complicated tasks. For this reason, Firebase was chosen as the BaaS[10] solution for this project. This service allows the developers to primarily focus on developing the front-end part of the application and provides APIs and SDKs for simplified back-end integration and communication.

The internal structure of the app was divided into three main parts:

1. **Landing page:** First page the user sees. It contains instructions, acknowledgments, and a non-mandatory name field.

2. **Survey:** Set of pages responsible for presenting the survey contents to the user and allowing them to rate individual image samples (Figure 6.1).

3. **Finish screen:** Page responsible for sending the collected survey data and acknowledging the user about the state of the transaction.

Each part was implemented as an independent component. Due to the simplicity of the app, all components belong to a single encapsulating module.

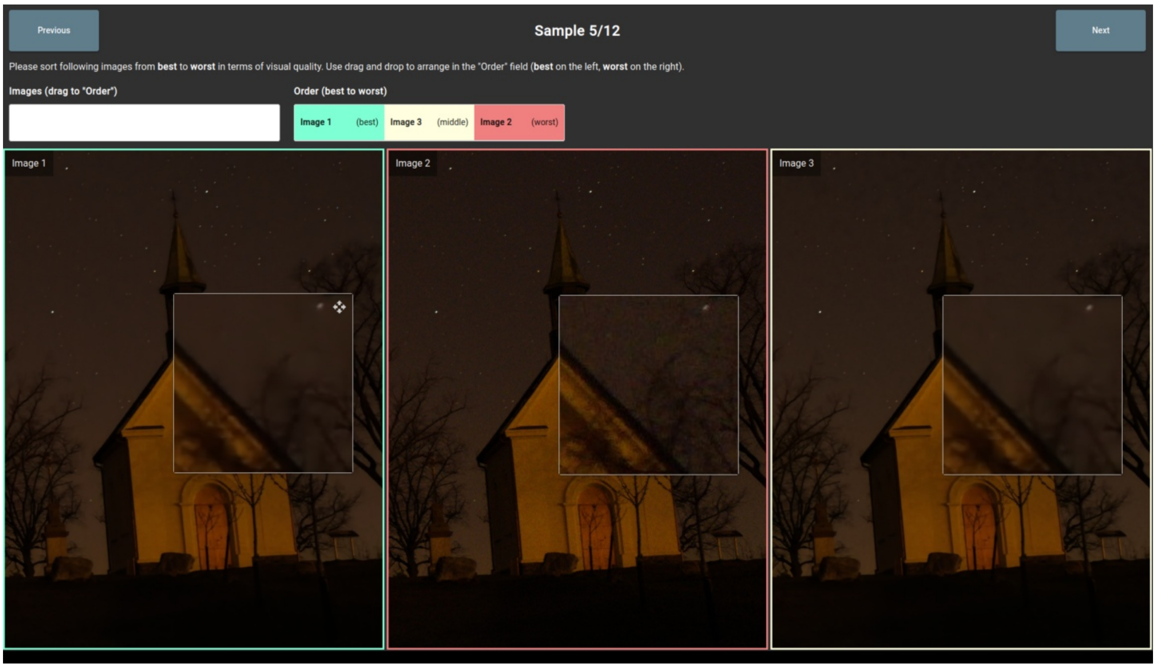
The survey consists of 12 image samples. Each sample contains three image variations: An original image from the Survey Dataset (Section 5.2), an image with reduced noise by BRDNet, and an image with reduced noise by DnCNN in randomized order. The user is required to rate the images in terms of visual quality. This is done by ordering the image labels from best to worst in a separate drag-and-drop area. Additionally, the user can inspect closer details by using a draggable magnifying element which provides 4× zoom on all of the image variations to help with spotting subtle differences between the presented images. The user can proceed only after successfully rating the active image sample. They also have the ability to inspect previous samples and/or change their ratings retrospectively if the user changes their preferences.

The user interface portion of this app was kept as simple as possible to avoid user distraction and keep the user's focus on the main task. However, a UI library was used to

[9] https://angular.io/

[10] BaaS – Back-end as a Service

aid with the implementation of the otherwise hard-to-implement features such as drag-and-drop areas and draggable elements. The library of choice was Angular Material[11] due to its ease of use, seamless integration with the Angular framework, and previous familiarity with using it.



Please sort following images from **best** to **worst** in terms of visual quality. Use drag and drop to arrange in the "Order" field (**best** on the left, **worst** on the right).

Images (drag to "Order")

Order (best to worst): Image 1 (best) Image 3 (middle) Image 2 (worst)

Figure 6.1: Demonstration of the survey user interface during the rating phase.

All image samples are stored in the Firebase Cloud Storage[12]. The images with reduced noise were saved in lossless compression format `png` to eliminate the influence of lossy compression on the final choice. The original images were all captured and saved using the lossy compression format `jpeg` and thus had to be saved in the same format. More information about the used dataset can be found in Section 5.2. The collected user data, that is the survey results, the user name (if given), and timestamp, are sent at the end of the survey as a JSON object to the Firebase Cloud Firestore[13] database for later evaluation. The whole application is hosted on the Firebase Hosting[14] platform to offer convenient access from the world wide web to the public.

At the time of writing this thesis, the application is hosted on the following url[15]. However, this hosting is not permanent and may be subject to change in the future. In such case, please refer to the source codes described in the Appendix A to perform local hosting.

[11] https://material.angular.io/

[12] https://firebase.google.com/docs/storage/

[13] https://firebase.google.com/docs/firestore/

[14] https://firebase.google.com/docs/hosting/

[15] https://denoise-survey.web.app/

Chapter 7

Experiments and results

This chapter describes all experiments that were carried out and presents their results. The goal of these experiments is to validate if the implementation of the two studied methods was proper, compare their relative performance in various situations, and to explore testing results that were missing in the original papers. Lastly, the run time performance of each trained method was measured and compared to provide additional data to base conclusions upon.

As was presented in Chapter 4, the two studied methods, DnCNN and BRDNet, were trained for two image noise reduction tasks – the blind Additive White Gaussian Noise (AWGN) reduction and the real image noise reduction. To differentiate between the two tasks the methods were trained for, the following naming scheme will be used throughout the rest of the thesis: the methods trained on blind AWGN will be referenced as DnCNN-B and BRDNet-B, alternatively the methods trained on real noisy images will be referenced to as DnCNN-R and BRDNet-R.

For evaluation of the noise reduction performance two commonly used metrics were used – PSNR and SSIM. Visual results are presented in several figures to compare noise reduction performance on images with both AWGN and real noise. In Section 7.4 results from the Public Survey are presented and statistically analyzed in order to draw several conclusions about the relative noise reduction performance of the studied methods on real noisy images.

All experiments were implemented in the Linux Mint 20.1 and Python 3.8.10 environments and ran on a PC with Intel(R) Core(TM) i5-4460 CPU, Nvidia GeForce GTX 1650 GPU, and 12 GB of DDR3 RAM.

7.1 Gaussian noise reduction

In the first round of experiments the two selected methods, DnCNN and BRDNet, were trained and tested in the task of removing Additive White Gaussian Noise. The goal of these experiments is to compare the relative performance of the two methods and verify if the implementation and training were proper by comparing the results with results proposed by the methods' authors. In the next section, models trained on AWGN will be used to determine the effect a real image dataset has on real image denoising performance.

In this section, the term *noise level* refers to the σ value of the Gaussian distribution of the noise that is added to the pixel values of the noiseless image in order to generate a noisy image. The pixel values of the image are represented as 8-bit values in 3 channels in the range of $[0, 255]$. If the values fall out of range with the addition of noise, the values are clipped.

Training

For training, 200 images from the BSD300 [29] dataset were used. Each image was randomly cropped to patches of size 50×50 px and then randomly rotated and/or flipped during mini-batch training. The Gaussian noise level range was set to $\sigma \in [0, 50]$ with $\mu = 0$. The reason for choosing random levels of σ is the presumption that the noise level present in digital photography is of random nature and thus a random level of image noise seems appropriate. It is noted that levels of σ as high as 75 were used in training on known fixed levels in the BRDNet publication [39]. However, such high levels of noise were deemed to be unrealistic for noise reduction in photos and thus were not considered in the training process. The mini-batch sizes were set to 64 and 20 for DnCNN-B and BRDNet-B respectively. Both models were trained over 100000 epochs to match the training volume presented in the papers. The training times for RDNet-B and DnCNN-B were 65 and 31 hours.

The loss function and optimizers were set similarly for both methods in accordance with the official implementations. The loss function used was the L1 loss, which utilizes the mean absolute error (MAE) to calculate the loss value. The optimizer of choice was Adam [25], a popular optimizer that is computationally efficient and leads to optimal results in the cases of both DnCNN and BRDNet. The learning rate was fixed to $1e - 4$.

Testing results

Testing was conducted on the CBSD68 and Kodak24 datasets (more information about chosen datasets can be found in Section 4.3). Noise reduction performance was measured by the PSNR and SSIM metrics on four different noise levels of 15, 25, 35, and 50. The results can be seen in Table 7.1. Results contain calculated metrics for the noisy image, DnCNN-B output image, and BRDNet-B output image in relation to the original image.

When observing the results, it is apparent that the BRDNet-B method is superior to the DnCNN-B method across all noise levels in both PSNR and SSIM metrics. The average difference in favor of BRDNet-B is 0.23 dB and 0.0053 for PSNR and SSIM respectively across both testing datasets and noise levels. The average improvement over the noisy image for DnCNN-B was 11.99 dB in PSNR and 0.5046 in SSIM while BRDNet-B showed an improvement of 12.23 dB in PSNR and 0.5099 in SSIM. These results are not surprising due to the fact that BRDNet is a more recent and complex network than DnCNN.

The results shown in table 7.1 correlate with results presented in the original BRDNet and DnCNN publications [39][47] and an independent overview of current noise reduction methods [38]. It needs to be noted that the original BRDNet publication presented results only for networks trained on fixed noise levels. However, the authors failed to mention this vital information when presenting their noise reduction results. Methods used in this thesis were trained on an unknown level of noise and thus their results perform marginally worse than in the original publication. The difference is approximately 0.2 dB in PSNR on average, which is very impressive given the more flexible nature of this approach. The overall testing results show that the training on Gaussian noise was done properly and successfully.

Visual results are presented in Figure 7.1 and Figure 7.2 on selected sample images from the Kodak24 and CBSD68 datasets. The difference between the noisy image and images denoised by both DnCNN-B and BRDNet-B is glaringly apparent – there can be

Datasets	Methods	$\sigma = 15$	$\sigma = 25$	$\sigma = 35$	$\sigma = 50$
CBSD68	–	24.61/0.5876	20.17/0.4150	17.25/0.3113	14.15/0.2180
	DnCNN-B	33.85/0.9302	31.15/0.8832	29.49/0.8408	27.81/0.7869
	BRDNet-B	34.03/0.9329	31.35/0.8878	29.70/0.8471	28.04/0.7941
Kodak24	–	24.61/0.5265	20.17/0.3500	17.25/0.2531	14.15/0.1714
	DnCNN-B	34.57/0.9223	32.07/0.8781	30.48/0.8384	28.84/0.7888
	BRDNet-B	34.79/0.9252	32.32/0.8829	30.75/0.8449	29.14/0.7966

Table 7.1: Average results of the noisy images (–), DnCNN-B output images, and BRDNet-B output images on the CBSD68 and Kodak24 datasets with noise levels of 15, 25, 35, and 50. Results are shown as PNSR(dB)/SSIM. Best results are shown in red.

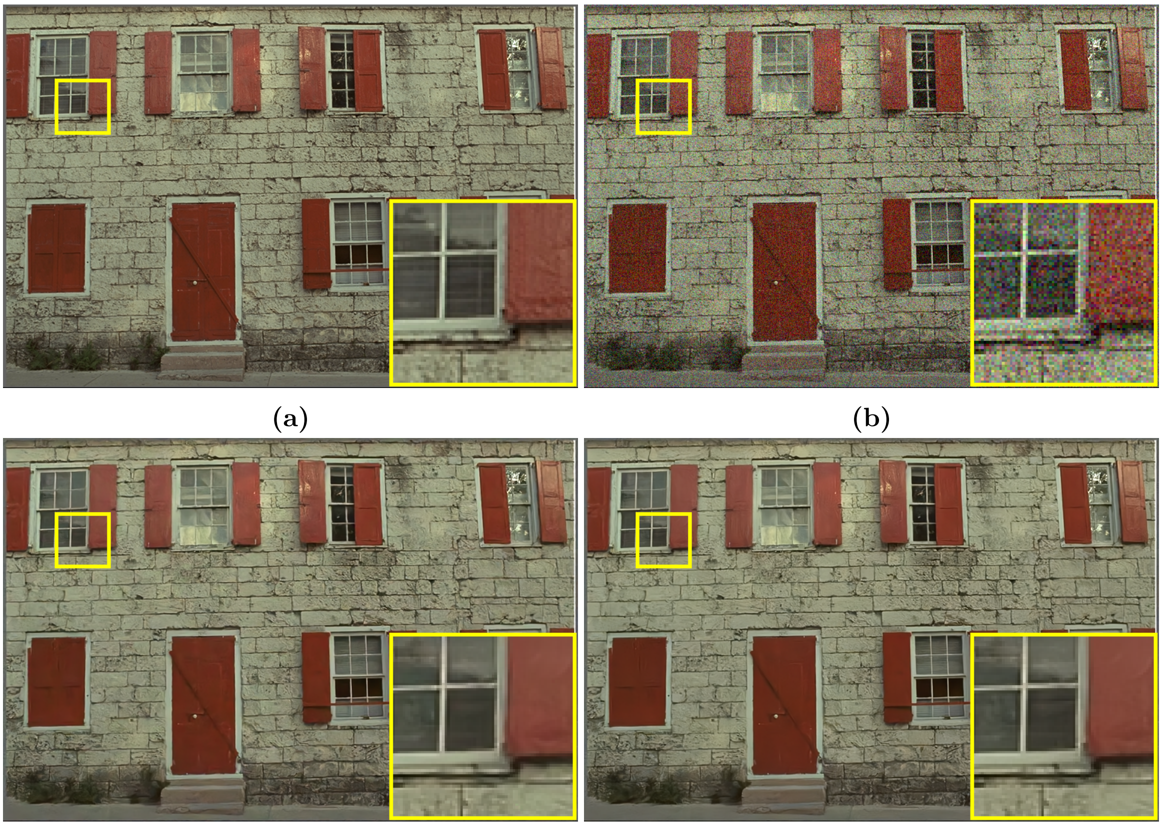


Figure 7.1: Gaussian noise reduction results from the Kodak24 dataset with $\sigma = 25$. Quality is measured as PSNR in relation to the original image. **(a)** Original image, **(b)** Noisy image/20.18 dB, **(c)** DnCNN-B/29.57 dB, **(d)** BRDNet-B/29.71 dB

no doubt that both of these methods produce state-of-the-art noise reduction performance in images with Gaussian noise. On the other hand, the visual difference between images produced by DnCNN-B and BRDNet-B is more subtle. Selected areas were enhanced in the figures demonstrating the noise reduction performance of both methods in order to aid with spotting the subtle differences. Upon closer inspection, the DnCNN-B appears to perform worse than BRDNet-B in preserving details in images, which is an expected result due to the fact that BRDNet was designed specifically to tackle this deficiency. This phenomenon

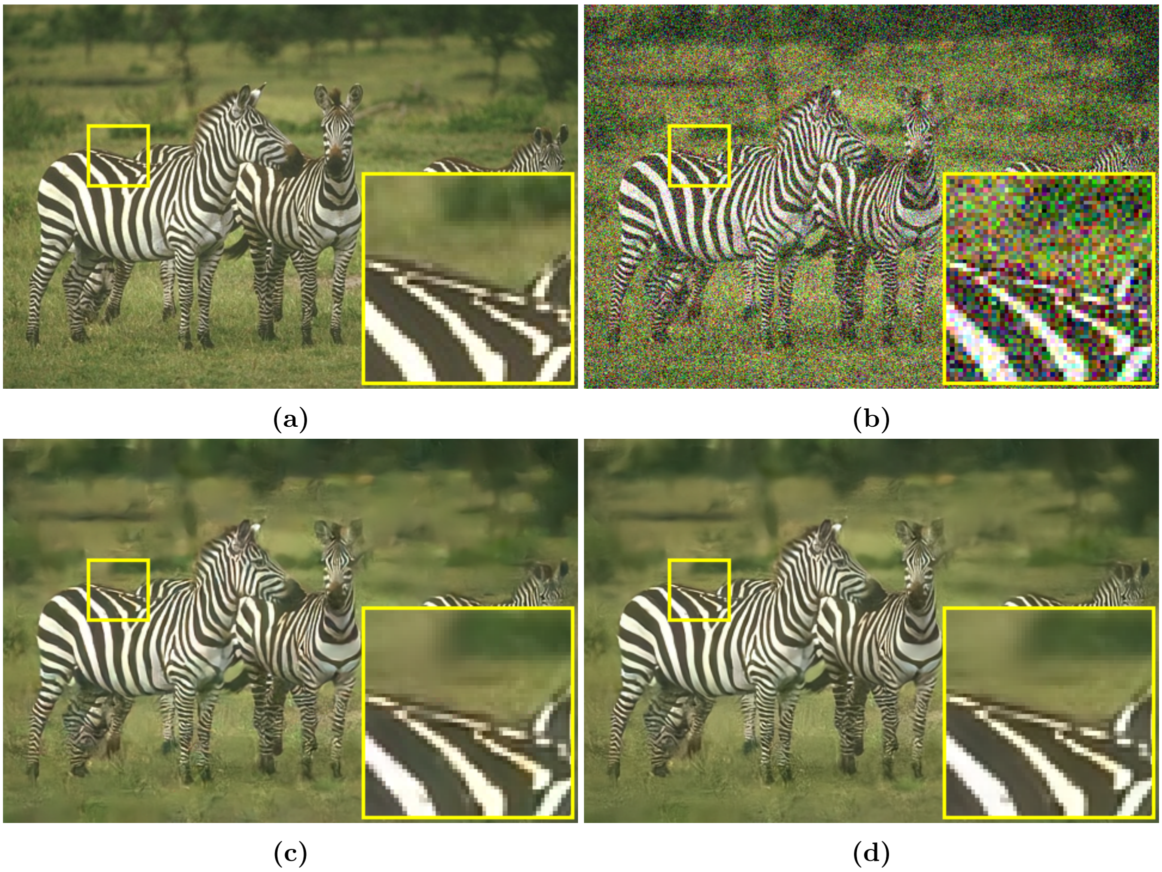


Figure 7.2: Gaussian noise reduction results from the CBSD68 dataset with $\sigma = 50$. Quality is measured as PSNR in relation to the original image. **(a)** Original image, **(b)** Noisy image/14.16 dB, **(c)** DnCNN-B/27.31 dB, **(d)** BRDNet-B/27.55 dB

can be clearly seen in Figure 7.1 where the window blinds are more distinct and textured in the BRDNet-B output image as opposed to the rather blurry output from DnCNN-B.

7.2 Real image noise reduction

In the second round of experiments, the two studied methods were trained and tested on real noisy image datasets to evaluate their denoising capabilities in photos and to determine which method offers better denoising performance in this task. The real image denoising performance of DnCNN-R and BRDNet-R was then compared with their counterparts trained on AWGN to establish the effects of a proper training dataset on real image denoising.

Training

For real image noise reduction, 100 images from the PolyU [44] dataset were used to train both methods. Each image was cropped into patches and augmented in the same way as in the random AWGN training process. The mini-batch sizes were set similarly to random Gaussian training to 64 and 20 for DnCNN-R and BRDNet-R respectively. The number of epochs was set to 150000 for both models. Training times were 50 and 24 hours for BRDNet-R and DnCNN-R respectively.

The loss function and optimizer were chosen similarly to the previous task – MAE and Adam were selected for the loss function and optimizer respectively. Both methods again utilized the same configuration in these two parameters and had the same fixed learning rate of $1e-4$.

Testing results

Testing of real image noise reduction was conducted on the CC dataset. The noise reduction performance was measured similarly to the Gaussian noise reduction in PSNR and SSIM metrics. The results were calculated for both the methods trained on random AWGN and methods trained on real noisy images in order to establish if training on real noisy datasets brings significant improvement in noise reduction performance.

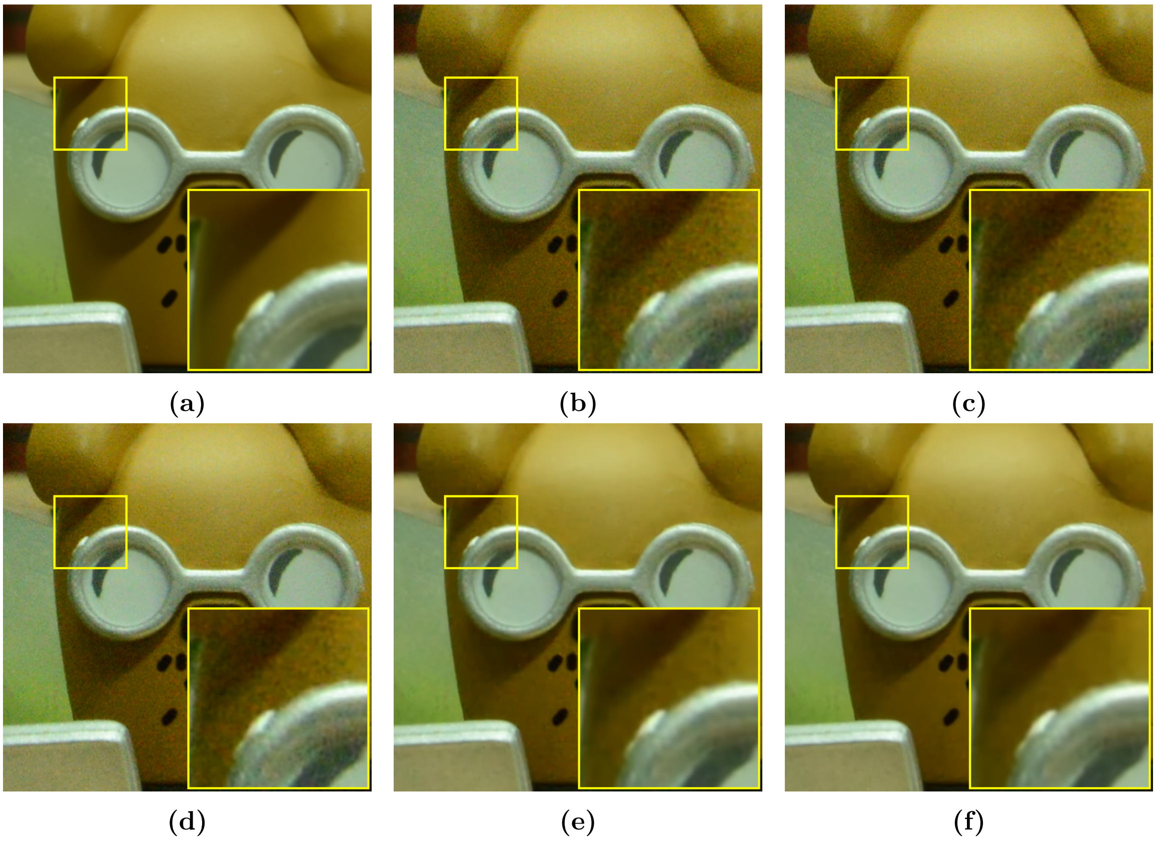
Dataset	Methods	PSNR (dB)	SSIM
CC	DnCNN-B	33.71	0.8224
	BRDNet-B	33.81	0.8274
	DnCNN-R	36.80	0.9346
	BRDNet-R	36.97	0.9374

Table 7.2: Average results of DnCNN-B, BRDNet-B, DnCNN-R, and BRDNet-R on the CC dataset consisting of real noisy images. Best results are shown in red.

The results shown in Table 7.2 show significant differences in performance between methods trained on random AWGN and methods trained on real noisy images – DnCNN-R improved by 3.09 dB in PSNR over its Gaussian noise alternative and BRDNet-R improved by 3.16 dB in PSNR. The visual comparison shown in Figure 7.3 clearly shows that DnCNN-B and BRDNet-B do not produce competitive results on images with real image noise.

When comparing DnCNN-R and BRDNet-R, similar observations as with DnCNN-B and BRDNet-B in synthetic noise testing can be made. The BRDNet-R takes the edge over the DnCNN-R in both the PSNR and SSIM metrics, which correlates with the expected results. The BRDNet-R again performs better than DnCNN-R in preserving fine image details as was the case in the Gaussian noise reduction testing. Figure 7.3 clearly shows BRDNet-R's superiority in this aspect.

When comparing results from Table 7.2 with results from BRDNet paper (Table 7.3), it is apparent that the BRDNet-R training was proper. However, we can notice that the authors list DnCNN's average performance on the CC dataset as 33.86 dB PSNR, which is significantly worse than what was observed on DnCNN-R in this thesis. The figure more closely resembles the results from DnCNN-B, which hints that the authors compared their proposed BRDNet method that was trained on real noisy images with DnCNN trained on AWGN. As was demonstrated, if DnCNN is trained on real noisy images, its performance on real image noise removal is orders of magnitude better. It is still outperformed by BRDNet, but the advantage is significantly smaller than what was presented in the original paper. It needs to be noted that no suggestions of malicious intent by the authors are being made because the DnCNN paper [47] itself did not present results on real noisy images. However, it is apparent that results presented in papers should be validated due to possibly misleading results leading to inaccurate and/or biased conclusions on the proposed method's performance.



(a) (b) (c)

(d) (e) (f)

Figure 7.3: Real image noise reduction results from the CC dataset. Quality is measured as PSNR in relation to the original image. **(a)** Original image, **(b)** Noisy image/33.26 dB, **(c)** DnCNN-B/34.05 dB, **(d)** BRDNet-B/34.26 dB, **(e)** DnCNN-R/38.70 dB, **(f)** BRDNet-R/39.18 dB

7.3 Overall results

The results from testing showed that the implementation of both methods was proper and their performance was on par with claims made in their respective papers. Testing also showed decisive results favoring the newer and more complex BRDNet over its counterpart DnCNN. The improvement was decisive in both AWGN and real image testing in both PSNR and SSIM metrics. The newer and wider architecture of BRDNet allows it to capture more detail and produce fewer artifacts which resulted in a more desirable output in all tested scenarios.

The real image noise reduction testing demonstrated the importance of a proper training dataset on the real image denoising performance. The methods trained on real images showed an improvement of approximately 3 dB PSNR and 0.11 SSIM over their counterparts trained on AWGN. These results also unveiled some discrepancies in the BRDNet publication's claims about the performance of DnCNN on real image denoising. It can be concluded that the BRDNet's wider and newer network architecture translates to better results in real image denoising as well with noticeable improvements over DnCNN in both PSNR and SSIM.

Camera settings	CBM3D	DnCNN	BRDNet
Canon 5D ISO=3200	39.76	37.26	37.63
	36.40	34.13	37.28
	36.37	34.09	37.75
Nikon D600 ISO=3200	34.18	33.62	34.55
	35.07	34.48	35.99
	37.13	35.41	38.62
Nikon D800 ISO=1600	36.81	35.79	39.22
	37.76	36.08	39.67
	37.51	35.48	39.04
Nikon D800 ISO=3200	35.05	34.08	38.28
	34.07	33.70	37.18
	34.42	33.31	38.85
Nikon D800 ISO=6400	31.13	29.83	32.75
	31.22	30.55	33.24
	30.97	30.09	32.89
Average	35.19	33.86	36.76

Table 7.3: Table taken from [39] comparing real image noise reduction performance of BRDNet and competing methods on the CC dataset. Only showing selected methods. Best results are shown in red and second-best are shown in blue. Shown values are in PSNR (dB).

Run times

In addition to output visual quality, another important aspect of a noise reduction method is its testing speed. Table 7.4 shows the run times of the studied methods on images with sizes of 512 × 512, 1024 × 1024, and 2048 × 2048. It needs to be noted that the results are bound to the tested hardware and should be used only for illustration purposes. Each method was tested on GPU by using the Pytorch Profiler[1] tool to measure the performance of the studied methods. The time for memory transfer between CPU and GPU is ignored as in [47][9]. The times were averaged over 100 iterations to provide more accurate results.

Methods	DnCNN-B	BRDNet-B	DnCNN-R	BRDNet-R
512 × 512	166 ms	355 ms	168 ms	359 ms
1024 × 1024	675 ms	1470 ms	673 ms	1450 ms
2048 × 2048	2682 ms	5780 ms	2712 ms	5850 ms

Table 7.4: Measured run times of the implemented studied methods.

The experimental results reveal that there is no significant difference in run time between similar methods trained on Gaussian noise and real image noise, which is to be expected and indicates that these models were trained and implemented properly. Additionally, the BRDNet method is on average 2.15× slower in processing given input than DnCNN. This result is to be expected due to BRDNet having a more complex network architecture than DnCNN. However, this difference in time complexity is quite significant and needs to be taken into account when comparing these methods.

[1] https://pytorch.org/docs/stable/profiler.html

7.4 Public survey results

The public survey was conducted for the purpose of evaluating the perceived visual quality of a real noisy image by the average person. The results were gathered in total from 41 respondents. As described in Section 6.3, the respondents were tasked with rating 12 image samples from the Survey Dataset. Each sample contained 3 versions of the same image: the original image, the image processed by DnCNN-R, and the image processed by BRDNet-R. In total, 492 image samples were rated. The results were then collected and analyzed in the Python language by using the Pandas[2] library to perform data analysis, the Seaborn[3] library to visualize the results and the SciPy[4] library to perform statistical tests. To help evaluate the results a scoring system for image samples was introduced. The user's ranking order for each sample was translated to scores in the following way: the best-rated image was given 2 points, the average 1 point, and the worst was given 0 points.

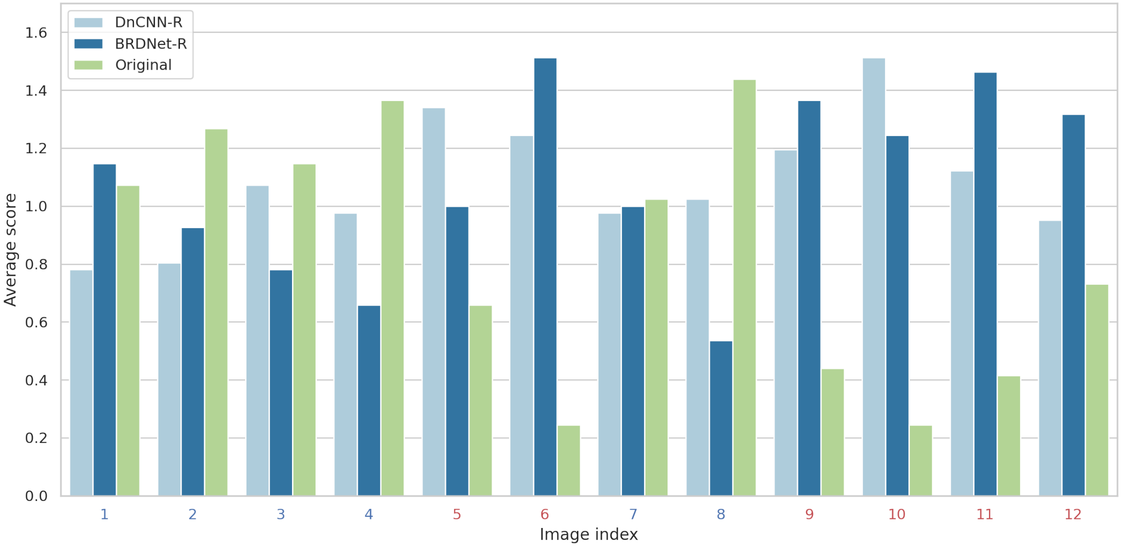


Figure 7.4: Bar plot showing the mean score values for each image sample from the public survey. A higher score is better. Indices of images with low image noise are shown in blue and images with high image noise are shown in red.

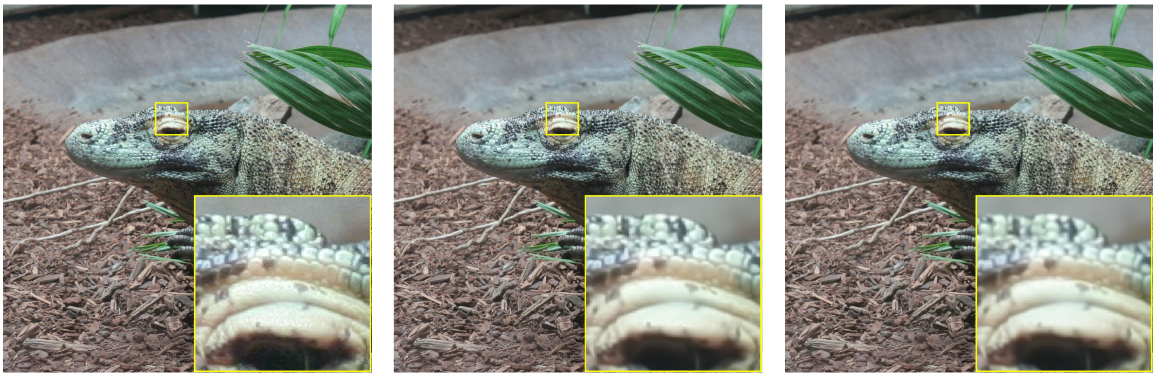
Firstly, the scores for each sample across all respondent entries were averaged and visualized in Figure 7.4. The results show strongly heterogenic results that show a strong preference for the original image over the images produced by the studied noise reduction methods in some of the image samples from the Survey Dataset. Specifically, the images numbered $\{2, 4, 8\}$ show a strong preference for the original image, images $\{1, 3, 7\}$ show indecisive results between the original and the versions with reduced noise, and images $\{5, 6, 8, 9, 10, 11, 12\}$ show decisive results in favor of the images with reduced noise by the DnCNN-R and BRDNet-R. After examination of the characteristics of the images in the Survey Dataset, a correlation between the image noise level and perceived visual quality produced by the noise reduction methods appears to exist. Image samples showing either a strong preference for the original image or indecisive results all share a common char-

[2] https://pandas.pydata.org/

[3] https://seaborn.pydata.org/

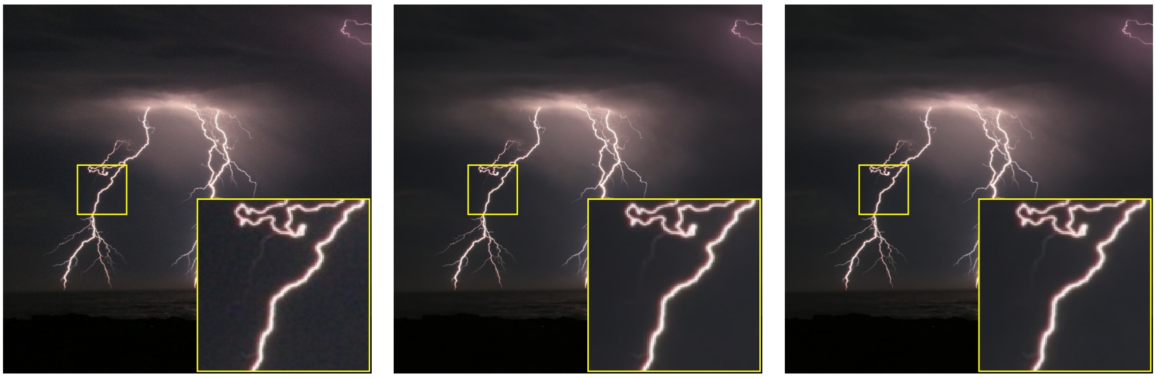
[4] https://scipy.org/

acteristic – lower levels of image noise. An example of this phenomenon can be seen in Figure 7.5 which shows image number 8 from the Survey Dataset which achieved one of the highest average scores for an original image in the survey. It is apparent that while reducing the unwanted image noise that is corrupting the image, the noise reduction methods visually appear to blur the image and soften or completely remove high-frequency details or textures from the original image. This trade-off between reduction of noise and image detail loss was perceived as unfavorable by the majority of respondents. On the other hand, image samples showing decisive results in favor of images with reduced noise all contain higher image noise caused by worse lighting conditions and higher ISO values. Figure 7.6 shows image number 10 from the Survey Dataset which contains a high level of image noise caused by poor lighting conditions, higher ISO level, and smaller camera sensor size. In these cases, the decrease in corruption caused by the higher image noise level seems to justify the trade-off between noise reduction and loss of image detail.



(a) **(b)** **(c)**

Figure 7.5: Noise reduction results on the image number 8 from the Survey Dataset with a low level of image noise: **(a)** Original image, **(b)** DnCNN-R, **(c)** BRDNet-R.



(a) **(b)** **(c)**

Figure 7.6: Noise reduction results on the image number 10 from the Survey Dataset with a high level of image noise: **(a)** Original image, **(b)** DnCNN-R, **(c)** BRDNet-R.

To determine if our assumptions that the studied methods perform worse in images with low image noise were correct, the following null, and alternative hypotheses were constructed:

Hypothesis 1 (H_0) *DnCNN-R and BRDNet-R perform equally well on images with low image noise and high image noise.*

Hypothesis 2 (H_1) *DnCNN-R and BRDNet-R* ***do not*** *perform equally well on images with low image noise and high image noise.*

To test these hypotheses we use the Person's chi-squared test[5] χ^2 which will help us determine if the null hypothesis can be rejected on a significance level of $\alpha = 95\%$. To apply this test, the image samples had to be categorized as either low-noise images or high-noise images in order to test the hypothesis. Images numbered $\{1, 2, 3, 4, 7, 8\}$ from the Survey Dataset were labeled as low-noise images and the remaining images numbered $\{5, 6, 9, 10, 11, 12\}$ were labeled as high-noise images as stated in Section 5.2. Additionally, conditions for evaluating if DnCNN-R and BRDNet-R improved the observed image had to be set. If the original image was selected as the best image (i.e. received full 2 points), the image sample was deemed as not improved by the noise reduction methods. Conversely, when the original image was selected as the worst image (i.e. received 0 points), the image was deemed to be improved by the noise reduction methods. However, in 5.69% of cases, the original image was ranked as the average (i.e. received 1 point) – in this case, the result was marked as not improved by the noise reduction methods. It needs to be stated that even if the result was marked as an improvement, the end result was not affected due to the low frequency of this phenomenon. Figure 7.7 visualises the total number of score occurrences for each method.

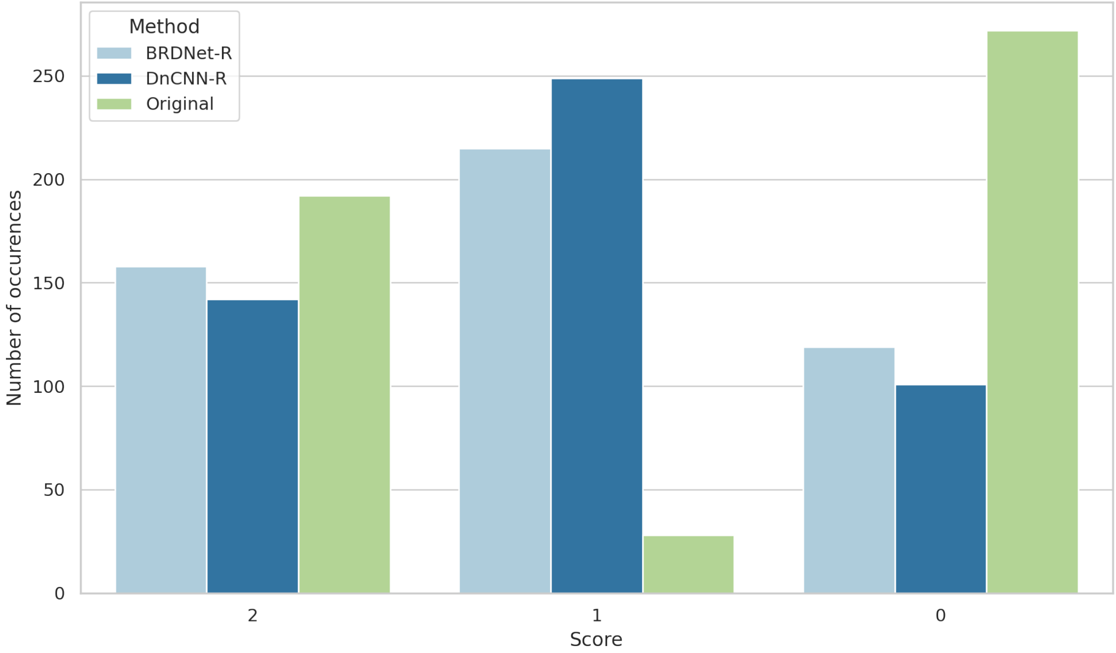


Figure 7.7: Bar plot showing the number of occurrences of score values for BRDNet-R, DnCNN-R and the original image collected from the public survey.

After applying the test and calculating the significance level, we observe that the test statistic exceeded the χ^2 critical value and thus the null hypothesis (Hypothesis 1) can be

[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.contingency.chi2_contingency.html

rejected and the alternative hypothesis (Hypothesis 2) can be accepted with the significance level of $\alpha = 95\%$. Moreover, after comparing the expected values from the χ^2 test with the gathered data, we can conclude that the DnCNN-R and BRDNet-R methods performed *worse* on images with low image noise, which corresponds with the predictions and visual results.

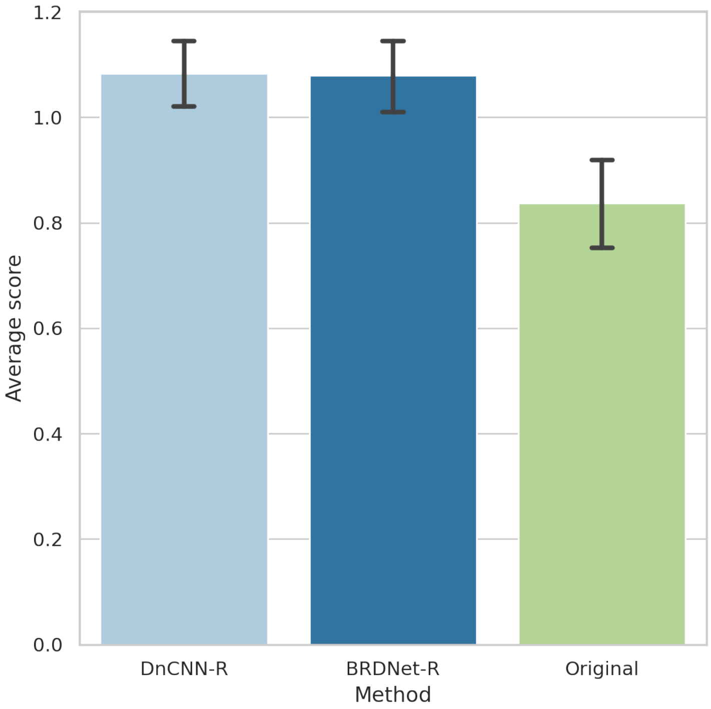


Figure 7.8: Bar plot showing the average scores and standard errors of the two noise reduction methods and the original image from the public survey results.

Lastly, the data gathered was used to analyze if the BRDNet-R method shows improvement over the DnCNN-R method as was the case in PSNR and SSIM metrics in Sections 7.1 and 7.2. For this purpose, mean scores for each method and the original image were computed and visualized in Figure 7.8. The mean score values and standard error values were 1.0833 ± 0.0314 for DnCNN-R, 1.0792 ± 0.0336 for BRDNet-R and 0.8373 ± 0.0432 for the original image. The average difference between the DnCNN-R and BRDNet-R was 0.0041 in favor of DnCNN-R. However, due to the high levels of standard error, the advantage of DnCNN-R over the BRDNet-R cannot be declared statistically significant. From these results, we can conclude that both DnCNN-R and BRDNet-R on average show improvement in perceived image quality in real noisy images. However, results show that the average person could not notice a significant difference in perceived visual quality between DnCNN-R and BRDNet-R.

7.5 Summary

The two studied noise reduction methods DnCNN and BRDNet were successfully trained and tested for both random AWGN reduction and real image noise reduction tasks. The BRDNet method outperformed the DnCNN method in all experiments solidifying its superiority in image denoising tasks due to its more advanced architecture. The BRDNet's ability to better preserve image details was demonstrated in several presented figures.

Real image noise testing revealed that the methods trained on blind AWGN were severely outperformed by their counterparts trained on real noisy images in both PSNR and SSIM metrics. The presumption that AWGN is unsuitable for modeling real image noise was thus proved. Furthermore, this highlighted the importance of a proper training dataset on the final denoising performance, especially when dealing with real image denoising.

Finally, the public survey revealed that conventional metrics used to measure image quality do not always correlate with perceived visual quality in photos. By statistically analyzing the gathered data, it was proven that DnCNN-R and BRDNet-R perform worse on images with low image noise and perform better on images with high levels of image noise. It was also proven that images with low image noise showed on average a decrease in perceived visual quality for both DnCNN-R and BRDNet-R with the original image being preferred by the respondents, which would be impossible to predict by using traditional metrics. Moreover, the advantage of BRDNet-R over the DnCNN-R in PSNR and SSIM did not translate to an advantage in perceived visual quality – both methods showed similar results within the measured margin of error in noise reduction performance.

7.6 Future work

Although it was proven that noise reduction methods based on deep convolutional neural networks deliver state-of-the-art performance, there is still a lot of room for improvement mainly in the detail preservation aspect. As was demonstrated in the carried out experiments, the loss of image detail can in certain cases lead to a decrease in perceived visual quality, which is an issue that should not be overlooked. Some possible solutions may include different network architectures or the incorporation of better-suited loss functions for the problem of removal of image noise in digital photography.

Experiments carried out in this thesis also showed the importance of training datasets on real image noise reduction performance. However, currently, the amount of real noisy image datasets is fairly limited. In my opinion, more emphasis should be given to constructing real noisy image datasets with greater variability of image sensors, scene types, and levels of image noise to facilitate greater flexibility and noise reduction performance in trained methods. In recent years, methods focused on synthesizing training datasets imitating the properties of real image noise have been proposed [46]. This approach seems promising due to the fact that big volumes of diverse data can be produced almost instantly. However, more testing and validation is still needed in this area as the comparison with real noisy image datasets is yet to be presented by the authors.

The results from the public survey also showed that the traditional evaluation metrics do not have to perfectly correlate with the perceived visual quality of digital images. More emphasis should be put on discovering new methods of evaluating digital image quality because it is apparent that the currently used metrics are imperfect.

Chapter 8

Conclusion

The goal of this thesis was to get acquainted with the problem of image denoising in digital photography and current methods based on deep neural networks. Two methods called DnCNN and BRDNet were selected to demonstrate the current state of denoising techniques.

The selected methods were successfully implemented and trained on publicly available datasets. The training was performed on both additive white Gaussian noise and real image noise to determine the influence of a proper training dataset on denoising performance in real photographs. Results have shown that models trained on real images showed improvement of over 3 dB in PSNR and 0.11 in SSIM over their counterparts trained on additive white Gaussian noise, which proved the necessity of proper training data in this task. Additional experiments highlighted the advantage of BRDNet over DnCNN in denoising performance due to its novel architecture. Evaluation on both real and synthetic data showed that BRDNet outperformed DnCNN in both PSNR and SSIM due to its ability to better preserve image details albeit for the cost of roughly double the computational time.

Additionally, a user testing experiment was designed, implemented, and carried out to find out if traditional metrics successfully represent the perceived visual quality of images. Results on the custom dataset have shown that statistically, both denoising methods perform worse in low-noise images and better in high-noise images. The tests revealed that, surprisingly, humans on average preferred the original noisy image in low-noise conditions due to greater detail fidelity. Moreover, results also revealed that the test subjects did not notice any significant difference in performance between DnCNN and BRDNet unlike in numerical testing, which proved the importance of appropriate quality measuring metrics in image denoising.

To summarize, I researched the field of image denoising by deep neural networks and selected two different methods to highlight the advancements in the field. I then successfully implemented both of them and performed extensive experiments to draw several conclusions about their performance and the importance of proper training datasets. I then developed a user testing tool along with an appropriate dataset to further evaluate the denoising performances of the two methods, which revealed interesting results regarding perceived image quality. Finally, conclusions were drawn with respect to the observed results, and ideas for future work were presented.

Bibliography

[1] Abdelhamed, A., Lin, S. and Brown, M. S. A High-Quality Denoising Dataset for Smartphone Cameras. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* June 2018.

[2] Anaya, J. and Barbu, A. RENOIR – A dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation.* 2018, vol. 51, p. 144–154. DOI: https://doi.org/10.1016/j.jvcir.2018.01.012. ISSN 1047-3203. Available at: https://www.sciencedirect.com/science/article/pii/S1047320318300208.

[3] Azzari, L., Borges, L. R. and Foi, A. Modeling and Estimation of Signal-Dependent and Correlated Noise. In: Bertalmío, M., ed. *Denoising of Photographic Images and Video: Fundamentals, Open Challenges and New Trends.* Cham: Springer International Publishing, 2018, p. 1–36. DOI: 10.1007/978-3-319-96029-6_1. ISBN 978-3-319-96029-6. Available at: https://doi.org/10.1007/978-3-319-96029-6_1.

[4] Benesty, J., Chen, J. and Huang, Y. Study of the widely linear Wiener filter for noise reduction. In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing.* 2010, p. 205–208. DOI: 10.1109/ICASSP.2010.5496033.

[5] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. et al. *Language Models are Few-Shot Learners.* arXiv, 2020. DOI: 10.48550/ARXIV.2005.14165. Available at: https://arxiv.org/abs/2005.14165.

[6] Buades, A., Coll, B. and Morel, J.-M. A non-local algorithm for image denoising. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).* 2005, vol. 2, p. 60–65 vol. 2. DOI: 10.1109/CVPR.2005.38.

[7] Chen, J., Chen, J., Chao, H. and Yang, M. Image Blind Denoising with Generative Adversarial Network Based Noise Modeling. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2018, p. 3155–3164.

[8] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O. et al. *Evaluating Large Language Models Trained on Code.* arXiv, 2021. DOI: 10.48550/ARXIV.2107.03374. Available at: https://arxiv.org/abs/2107.03374.

[9] Chen, Y. and Pock, T. Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2017, vol. 39, no. 6, p. 1256–1272. DOI: 10.1109/TPAMI.2016.2596743.

[10] Couturier, R., Perrot, G. and Salomon, M. Image Denoising Using a Deep Encoder-Decoder Network with Skip Connections. In: Cheng, L., Leung, A. C. S. and Ozawa, S., ed. *Neural Information Processing*. Cham: Springer International Publishing, 2018, p. 554–565. ISBN 978-3-030-04224-0.

[11] Dabov, K., Foi, A., Katkovnik, V. and Egiazarian, K. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*. 2007, vol. 16, no. 8, p. 2080–2095. DOI: 10.1109/TIP.2007.901238.

[12] Delon, J. and Houdard, A. Gaussian Priors for Image Denoising. In: Bertalmío, M., ed. *Denoising of Photographic Images and Video: Fundamentals, Open Challenges and New Trends*. Cham: Springer International Publishing, 2018, p. 125–149. DOI: 10.1007/978-3-319-96029-6_5. ISBN 978-3-319-96029-6. Available at: https://doi.org/10.1007/978-3-319-96029-6_5.

[13] Fan, L., Li, X., Guo, Q. and Zhang, C. Nonlocal image denoising using edge-based similarity metric and adaptive parameter selection. *Science China Information Sciences*. Jan 2018, vol. 61, no. 4, p. 049101. DOI: 10.1007/s11432-017-9207-9. ISSN 1869-1919. Available at: https://doi.org/10.1007/s11432-017-9207-9.

[14] Fan, L., Zhang, F., Fan, H. and Zhang, C. Brief review of image denoising techniques. *Visual computing for industry, biomedicine, and art*. Springer Singapore. Jul 2019, vol. 2, no. 1, p. 7–7. DOI: 10.1186/s42492-019-0016-7. ISSN 2524-4442. 32240414[pmid]. Available at: https://pubmed.ncbi.nlm.nih.gov/32240414.

[15] Frazen, R. *Kodak lossless true color image suite* [online]. 1999 [cit. 2022-04-10]. Available at: http://r0k.us/graphics/kodak/.

[16] Gatcum, C. *Kompletní fotografie: nejlepší fotografie z každého aparátu*. Vydání prvníth ed. Brno: Zoner Press, 2018. ISBN 978-80-7413-378-7.

[17] Gu, S., Xie, Q., Meng, D., Zuo, W., Feng, X. et al. Weighted Nuclear Norm Minimization and Its Applications to Low Level Vision. *International Journal of Computer Vision*. Jan 2017, vol. 121, no. 2, p. 183–208. DOI: 10.1007/s11263-016-0930-5. ISSN 1573-1405. Available at: https://doi.org/10.1007/s11263-016-0930-5.

[18] Guo, S., Yan, Z., Zhang, K., Zuo, W. and Zhang, L. Toward Convolutional Blind Denoising of Real Photographs. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, p. 1712–1722. DOI: 10.1109/CVPR.2019.00181.

[19] He, K., Zhang, X., Ren, S. and Sun, J. *Deep Residual Learning for Image Recognition*. arXiv, 2015. DOI: 10.48550/ARXIV.1512.03385. Available at: https://arxiv.org/abs/1512.03385.

[20] Huang, J.-B., Singh, A. and Ahuja, N. Single image super-resolution from transformed self-exemplars. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 5197–5206. DOI: 10.1109/CVPR.2015.7299156.

[21] Ioffe, S. *Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models.* arXiv, 2017. DOI: 10.48550/ARXIV.1702.03275. Available at: https://arxiv.org/abs/1702.03275.

[22] Ioffe, S. and Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.* arXiv, 2015. DOI: 10.48550/ARXIV.1502.03167. Available at: https://arxiv.org/abs/1502.03167.

[23] Jain, V. and Seung, H. Natural Image Denoising with Convolutional Networks. In:. January 2008, p. 769–776.

[24] Johnson, J., Alahi, A. and Fei Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., ed. *Computer Vision – ECCV 2016.* Cham: Springer International Publishing, 2016, p. 694–711. ISBN 978-3-319-46475-6.

[25] Kingma, D. P. and Ba, J. *Adam: A Method for Stochastic Optimization.* arXiv, 2014. DOI: 10.48550/ARXIV.1412.6980. Available at: https://arxiv.org/abs/1412.6980.

[26] Ma, C., Yang, C.-Y., Yang, X. and Yang, M.-H. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding.* 2017, vol. 158, p. 1–16. DOI: https://doi.org/10.1016/j.cviu.2016.12.009. ISSN 1077-3142. Available at: https://www.sciencedirect.com/science/article/pii/S107731421630203X.

[27] Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H. et al. Waterloo Exploration Database: New Challenges for Image Quality Assessment Models. *IEEE Transactions on Image Processing.* Feb. 2017, vol. 26, no. 2, p. 1004–1016.

[28] Mahmoudi, M. and Sapiro, G. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters.* 2005, vol. 12, no. 12, p. 839–842. DOI: 10.1109/LSP.2005.859509.

[29] Martin, D., Fowlkes, C., Tal, D. and Malik, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: *Proc. 8th Int'l Conf. Computer Vision.* July 2001, vol. 2, p. 416–423.

[30] Martin, D., Fowlkes, C., Tal, D. and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.* 2001, vol. 2, p. 416–423 vol.2. DOI: 10.1109/ICCV.2001.937655.

[31] Mittal, A., Soundararajan, R. and Bovik, A. C. Making a "Completely Blind" Image Quality Analyzer. *IEEE Signal Processing Letters.* 2013, vol. 20, no. 3, p. 209–212. DOI: 10.1109/LSP.2012.2227726.

[32] Moeller, M. and Cremers, D. Image Denoising—Old and New. In: Bertalmío, M., ed. *Denoising of Photographic Images and Video: Fundamentals, Open Challenges and New Trends.* Cham: Springer International Publishing, 2018,

p. 63–91. DOI: 10.1007/978-3-319-96029-6_3. ISBN 978-3-319-96029-6. Available at: https://doi.org/10.1007/978-3-319-96029-6_3.

[33] N, V., D, P., Bh, M. C., Channappayya, S. S. and Medasani, S. S. Blind image quality evaluation using perception based features. In: *2015 Twenty First National Conference on Communications (NCC)*. 2015, p. 1–6. DOI: 10.1109/NCC.2015.7084843.

[34] Nam, S., Hwang, Y., Matsushita, Y. and Kim, S. J. A Holistic Approach to Cross-Channel Image Noise Modeling and its Application to Image Denoising. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[35] Plötz, T. and Roth, S. Benchmarking Denoising Algorithms with Real Photographs. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, p. 2750–2759. DOI: 10.1109/CVPR.2017.294.

[36] Rudin, L. I., Osher, S. and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*. 1992, vol. 60, no. 1, p. 259–268. DOI: https://doi.org/10.1016/0167-2789(92)90242-F. ISSN 0167-2789. Available at: https://www.sciencedirect.com/science/article/pii/016727899290242F.

[37] Simonyan, K. and Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv, 2014. DOI: 10.48550/ARXIV.1409.1556. Available at: https://arxiv.org/abs/1409.1556.

[38] Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W. et al. Deep learning on image denoising: An overview. *Neural Networks*. 2020, vol. 131, p. 251–275. DOI: https://doi.org/10.1016/j.neunet.2020.07.025. ISSN 0893-6080. Available at: https://www.sciencedirect.com/science/article/pii/S0893608020302665.

[39] Tian, C., Xu, Y. and Zuo, W. Image denoising using deep CNN with batch renormalization. *Neural Networks*. 2020, vol. 121, p. 461–473. DOI: https://doi.org/10.1016/j.neunet.2019.08.022. ISSN 0893-6080. Available at: https://www.sciencedirect.com/science/article/pii/S0893608019302394.

[40] Tian, C., Xu, Y., Zuo, W., Du, B., Lin, C.-W. et al. Designing and training of a dual CNN for image denoising. *Knowledge-Based Systems*. 2021, vol. 226, p. 106949. DOI: https://doi.org/10.1016/j.knosys.2021.106949. ISSN 0950-7051. Available at: https://www.sciencedirect.com/science/article/pii/S0950705121002124.

[41] Tomasi, C. and Manduchi, R. Bilateral filtering for gray and color images. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, p. 839–846. DOI: 10.1109/ICCV.1998.710815.

[42] Vansteenkiste, E., Weken, D. Van der, Philips, W. and Kerre, E. Perceived Image Quality Measurement of State-of-the-Art Noise Reduction Schemes. In: Blanc Talon, J., Philips, W., Popescu, D. and Scheunders, P., ed. *Advanced Concepts for Intelligent Vision Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, p. 114–126. ISBN 978-3-540-44632-3.

[43] Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*. 2004, vol. 13, no. 4, p. 600–612. DOI: 10.1109/TIP.2003.819861.

[44] Xu, J., Li, H., Liang, Z., Zhang, D. and Zhang, L. *Real-world Noisy Image Denoising: A New Benchmark*. arXiv, 2018. DOI: 10.48550/ARXIV.1804.02603. Available at: https://arxiv.org/abs/1804.02603.

[45] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H. et al. Deep Learning for Single Image Super-Resolution: A Brief Review. *IEEE Transactions on Multimedia*. 2019, vol. 21, no. 12, p. 3106–3121. DOI: 10.1109/TMM.2019.2919431.

[46] Zhang, K., Li, Y., Liang, J., Cao, J., Zhang, Y. et al. Practical Blind Denoising via Swin-Conv-UNet and Data Synthesis. *ArXiv preprint*. 2022.

[47] Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*. 2017, vol. 26, no. 7, p. 3142–3155. DOI: 10.1109/TIP.2017.2662206.

[48] Zhang, L., Wu, X., Buades, A. and Li, X. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging*. SPIE. 2011, vol. 20, no. 2, p. 1 – 17. DOI: 10.1117/1.3600632. Available at: https://doi.org/10.1117/1.3600632.

Appendix A

Contents of the included storage media

- `survey/` Folder with the Public Survey source code and additional data
- `methods/` Folder with methods' source codes and additional data
- `tech-report/` Folder with LaTeXrouce files
- `video.mp4` Demonstration video
- `README.md` Project description with setup instructions
- `tech-report.pdf` Technical report

Appendix B

Video

The video summarizing this thesis can be found at https://youtu.be/cyRqxUzki-c. The video is also present on the included storage media.