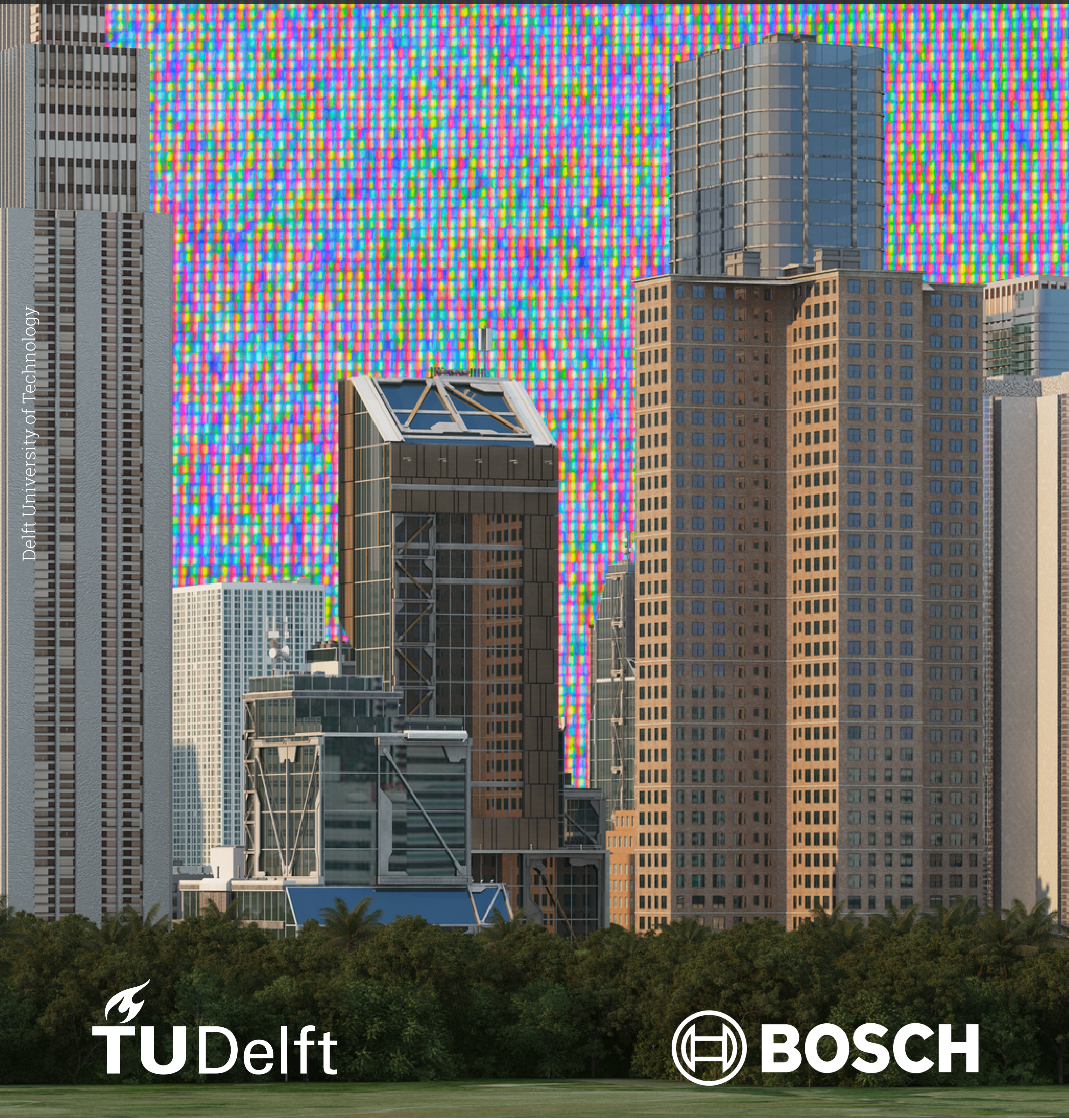
One model, denoise them all!

A Comprehensive Investigation of Denoising Transfer Learning

MSc Computer Science - AI track

Dajt Mullaj



Delft University of Technology

TU Delft

BOSCH

One model, denoise them all!

A Comprehensive Investigation of Denoising Transfer Learning

by

Dajt Mullaj

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Wednesday August 30, 2023 at 10:00 AM.

Supervisors:	Dr. O.S. Kayhan Dr. J.C. Van Gemert	
Project duration:	November 14, 2023 – August 30, 2023	
Thesis committee:	Dr. J.C. Van Gemert,	Associate Professor, TU Delft
	Prof.Dr. M.M. de Weerdt,	Full Professor, TU Delft
	Dr. O.S. Kayhan,	Bosch Security System, TU Delft

Preface

This thesis was conducted in collaboration with Bosch Security Systems B.V. and is part of my Master of Science at The Delft University of Technology.

I would like to sincerely thank my supervisor Osman. Our constant discussions and his support made it easy to keep feeling motivated and engaged throughout my research. I also extend my thanks to my peers Rick and Thomas, who were always available to discuss findings together. I then would like to thank Jan, who was always insightful when addressing results, and the team at Bosch Security System, for this opportunity and their guidance. Finally I would like to thank my parents for their constant and endless support.

Dajt Mullaj
Delft, August 2023

Contents

1
Introduction

Every imaging system is subject to noise. Noise emerges from several sources as low lit environments, short exposure times or small image sensors. Image noise removal is an important field of study across many sectors which rely on imaging quality, as surveillance systems and medical application. Deep learning methods leveraging Convolutional Neural Networks (CNNs) have shown excellent performance when applied to the denoising task [6, 9, 27]. However, CNNs are typically trained on extensive datasets, composed by clean-noisy image pairs. Since capturing a clean-noisy dataset using digital cameras is a tedious and complex task [6, 18], publicly available datasets with real-noise images are few and small. This limits the applicability of CNNs denoisers with real world imaging systems.

In this thesis we address data scarce denoising by employing transfer learning. During transfer learning a CNN is first pre-trained on a large dataset and then finetuned with a small target dataset. Researchers have shown that is possible, using transfer learning, to achieve competitive results on real-noise data, pre-training a CNN denoiser with numerous synthetic-noise samples [13]. Nevertheless transfer learning presents its own set of challenges. It is unclear how dataset size affects final denoising performance. Furthermore, finetuning all CNNs parameters with small datasets can lead to overfitting, deteriorating performance on real-noise images. Lastly, how to evaluate the domain discrepancy between the pre-training noise distribution and the finetuning noise distribution remains unanswered.

We set to investigate denoising transfer learning by establishing a precise scientific environment, where we define transfer learning using 4 key variables: (i) the size of the pre-training dataset; (ii) the size of the finetuning dataset; (iii) the set of CNN's parameters updated during finetuning; and (iv) the similarity between the pre-training and finetuning noise distributions. We aim to analyse the effect of each variable under progressively complex experimental set-ups and different testing conditions, to demonstrate generality of the uncovered concepts. We show our findings have direct applicability on real-noise images and, thus, further the conceptual understanding of denoising transfer learning.

This thesis is structured as follows. In Chapter 2 we present our findings and research methodology as a scientific article. At the end of the article we provide an appendix with additional results, aimed at validating applicability of our findings across different environments. Chapter 3 is intended for computer vision and deep learning non-experts. In Chapter 3.1 we provide background information for gaining familiarity with image noise concepts such as noise distribution models and metrics for noise quantification. In Chapter 3.2 we discuss deep learning and how we employ it in our research. Lastly Chapter 3.3 provides visual information regarding the image datasets used in our experiments.

2

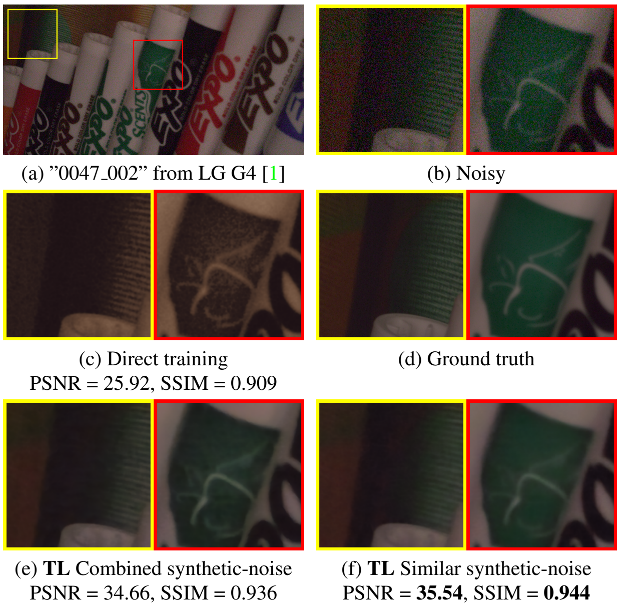
Scientific Article

One model, denoise them all!

Dajt Mullaj[1,2] Osman Semih Kayhan[1,2] Jan C.van Gemert[2]
[1] Bosch Security Systems [2] Delft University of Technology

Abstract

Deep convolutional neural networks (CNNs) have achieved current state-of-the-art in image denoising, but require large datasets for training. Their performance remains limited on smaller real-noise datasets. In this paper, we investigate robust deep learning denoising using transfer learning. We explore the impact of dataset sizes, CNN parameter updates, and noise distribution similarity. Our findings demonstrate that finetuning the decoder while fixing the encoder of an encoder-decoder network architecture avoids overfitting during transfer learning. Moreover, we introduce the concept of noise similarity in transfer learning, showing that reducing the similarity distance between pre-training and finetuning noise significantly enhances CNN performance in denoising. Our results are demonstrated on multiple datasets and various noise camera models. Finally, we validate the robustness and applicability of our approach on real-noise images. All our results and analyses hold and generalize across different datasets, highlighting our insights into the potential of transfer learning for image denoising.


(a) ”0047_002” from LG G4 [1]

(b) Noisy

(c) Direct training
PSNR = 25.92, SSIM = 0.909

(d) Ground truth

(e) **TL** Combined synthetic-noise
PSNR = 34.66, SSIM = 0.936

(f) **TL** Similar synthetic-noise
PSNR = **35.54**, SSIM = **0.944**

Figure 1. Comparing direct training and transfer learning (TL) approaches using real-noise images captured with the LG G4 camera from SSID-Small dataset [1]. With a small dataset size, typical CNN supervised training (c) fails to achieve satisfactory results. Moreover, pre-training with synthetic-noise similar to the target real-noise (f) achieves better results compared to pre-training with a generic collection of synthetic noise (e). This is clearly visible in the image, where color distortions and noisy artifacts are evident.

1. Introduction

Image denoising is a fundamental aspect of image processing and refers to the task of removing unwanted distortions from digital images to enhance their quality. Digital camera noise originates from multiple sources, such as small image sensor size, short exposure times, electronic interference, and the camera's image processing pipeline.

Recently image denosing deep learning methods using Convolutional Neural Networks (CNNs) have achieved state-of-the-art results [6, 17, 41, 42]. These CNN-based methods require extensive datasets consisting of clean-noisy image pairs and are typically trained with synthetic Gaussian noise. However, the performance of deep denoisers trained with Gaussian noise deteriorates considerably when confronted with real-noise images, due to the domain discrepancy between training-noise and real-noise [17, 22].

Addressing this issue by gathering large real-noise datasets captured using digital cameras is a demanding task [3, 11, 29]. It involves overcoming operational complexities, such as minimizing object motion during prolonged exposure, and post-processing challenges, as the generation of valid clean images to use for ground truth data. Due to these difficulties, current publicly available real-noise datasets generally present less than 40 distinct image scenes [3, 29, 38].

An alternative approach involves employing realistic noise distributions during synthetic-noise training. CBDNet [17] employs this technique to achieve good performance on real-noise images. Nevertheless, a domain discrepancy between the training and testing noise still exists.

To reduce the noise domain discrepancy, transfer learning has been employed to adapt a model pre-trained on a large dataset to a target dataset with a different noise distribution [9, 22, 27, 36]. AINDNet [22] is pre-trained on synthetic-noise data and finetuned on a smaller real-noise dataset to enhance denoising of real-world images. AINDNet achieves strong performance with several benchmarking real-noise datasets such as DND [28]. While AIDNet proposes an effective transfer learning scheme, it is specialized for its own architecture. Current transfer learning methods focus on the design of specific CNN architectures, which makes their general applicability limited.

Furthermore, despite its advantages, transfer learning presents its own set of challenges. The connection between performance and the sizes of pre-training or finetuning datasets remains undefined for image denosing. Moreover, adapting the entire complete parameter set of a CNN can lead to overfitting [26, 27]. Lastly, the choice of synthetic-noise for pre-training lacks clear criteria.

In this study, we investigate how to train a robust deep denoiser using transfer learning. We specifically define the denoising transfer learning framework with four key variables: (i) the size of the pre-training dataset; (ii) the size of the finetuning dataset; (iii) the set of CNN's parameters updated during finetuning; and (iv) the similarity between the pre-training and finetuning noise distributions.

We conduct experiments with varying pre-training and finetuning dataset size to explore the conditions under which our findings remain consistent and to assess how denoising performance responds to data availability. In the more realistic experimental scenarios, we maintain fixed dataset sizes to simulate real-world conditions.

We analyze which subset of the CNNs parameter needs to be updated, to prevent overfitting. We select the U-Net [31] network as our architecture framework. U-Net's efficient encoder-decoder structure, comprising a contracting and expanding part, is widely adopted in many state-of-the-art denoising networks [14, 17, 22]. Our empirical results demonstrate that finetuning the decoder part of the network outperforms finetuning the encoder. Furthermore, updating only the expanding part of the network effectively minimizes overfitting.

We develop an approach to compute noise distributions similarity, independent to the choice of the specific similarity distance metric. We show that pre-training a network using synthetic-noise similar to the finetuning noise effectively reduces domain discrepancy and outperforms a network pre-trained on less similar noise. Notably using similar noise for pre-training outperforms using a range of generic synthetic camera noise distributions (see Fig. 1).

We conduct our analysis using realistic synthetic-noise. We use the Poisson-Gaussian noise model [15], which is suitable for modeling real-noise [3, 15], to simulate different camera noise distributions. Next, we validate our results using camera-specific synthetic-noise models tailored to mimic the characteristics of distinct real cameras. Lastly we demonstrate the influence of noise similarity on real-noise datasets, SSID-Small [1] and RENOIR [3].

To sum up, this paper makes the following contributions:

- We present a comprehensive evaluation of transfer learning in the denoising context, including multiple noise models, dataset sizes and applicability to real noise scenarios.

- We identify which components of an encoder-decoder network architecture lead to the most significant improvements when updated, highlighting how updating only the decoder gives faster convergence and can prevent overfitting for small finetuning dataset.

- We introduce the concept of noise adaptation via transfer learning, by quantifying noise similarity and assessing its effect in the finetuning process.

2. Related work

Deep learning denoising. Since the introduction of DnCNN [41], a denoising CNN using batch normalization [20] and residual learning [18], CNN-based methods have outperformed previous state-of-the-art non-CNN denoising methods [5, 13, 16].

Notably, encoder-decoder structures with a contracting and expanding part, such as U-Net [31], have standardized deep learning denoising methods. This architecure is used in several networks which achieve state-of-the-art performance with real noise datasets, like CBDNet [17] and AINDNet [22]. The flexibility of the encoder-decoder structure is used in deep denoisers which exploit transformers, SUNnet [14], or more generally attention, EnlightenGAN [21]. Furthermore a U-Net architecture is effective at capturing captures essential low-level image priors [35]. From these observations we utilize U-Net for our framework architecture, when conducting our denoising experiments.

Transfer Learning in Denoising. Deep-CNN methods are typically trained with supervision, requiring large datasets of clean-noisy image pairs. Nevertheless, collecting real-noise image pairs using a digital camera is a challenging task. Hence the collected datasets are usually too small for typical supervised learning. To address the problem CBDNet [17] is trained using a combination of real-noise data and images corrupted by realistic synthetic noise. However, this approach can lead to training instability due to training with different camera-noise distributions and is not effective for training distinct real-noise datasets [22].

Transfer learning adaptation methods have shown promising results in scenarios where only a limited number of image pairs are available. AINDNet [22] adapts from synthetic-noise to real-noise by updating the normalization parameters of the model. Similarly, LIDIA [36], a lightweight architecture for image denoising, achieves state-of-the-art results by adapting the network to a target input image. While denoising transfer learning schemes are effective, they are usually constrained to a specific network architecture. In this study, we aim to define and explore transfer learning characteristics that can be utilized by general CNN architectures.

Parameters update. Finetuning the full parameter set of a CNN on a small dataset can lead to overfitting [27]. Due to their large number of parameters, CNNs can fit the finetuning noise-images without effectively capturing the underlying noise distribution patterns. AINDNet [22] prevents overfitting by only updating the normalization parameters. LIDIA [36] is a specialized architecture which employs a reduced number of parameters. More common techniques as early stopping [34], also show effectiveness in preventing overfitting during denoising transfer learning. Gain-Tuning [27] introduces a universal transfer learning scheme aimed at preventing overfitting by updating only a scaling factor preceding each layer. Recent research has examined finetuning different U-Net parameters [2, 12]. Results for the segmentation task show that finetuning the early layers is more effective than finetuning the final layers of a U-Net [2]. We investigate whether finetuning the decoder or the encoder part of U-Net is more important in denoising, and if fixing either effectively prevents overfitting.

Noise distribution similarity. Quantifying similarity between different distributions is a well-known topic with several techniques. Similarity distances, quantifying how far apart two distributions are, can be categorized as satisfying metric proprieties or not, in which case they are referred to as divergences [7]. Euclidean distance [7] can be used as a metric for comparing distributions, treating probability function values as points in a vector space. Common metrics rely on Shannon's concept of entropy [32]. Jensen-Shannon distance [24] is based on the Kullback–Leibler divergence [23], with the important difference that it is symmetric and always finite. Variation of information [33] uses mutual information and entropy to compute the distance between discrete probabilities distributions. A conceptually different metric is Wasserstein distance [30], measuring the ”cost” of turning one distribution into another.

Noise similarity is crucial in transfer learning methods for denoising, since it measures the discrepancy between the pre-training noise domain and the finetuning noise domain. We use similarity metrics to evaluate these discrepancies and show that pre-training on a noise distributions more similar to the finetuning noise produces better results.

Noise modelling. The noise in real images originates from multiple sources, as the camera sensor size or type and its high susceptibility to photon noise. The Poisson-Gaussian [15] noise model has been proposed to approximate the noise in real images. Poisson-Gaussian is composed by two parts: the Poisson component, which models photon sensing, and the Gaussian component, used to represent the remaining stationary disturbances in the image. The model is therefore signal-dependent and can accurately fit real image noise [3, 4, 15, 43]. Several techniques can be used to estimate the Poisson-Gaussian parameters given a set of real-noise images [4], using the variance of the values of the noisy image pair. We use the Poisson-Gaussian noise model to simulate different camera's noise distributions and to estimate the noise distribution of real-noise images. Furthermore, to validate our results we use synthetic-noise generated by camera-specific noise models and real-noise images captured by distinct cameras.

3. Transfer Learning within Denosing

In a typical supervised context, for a dataset $\mathcal{D}$ composed by clean-noisy image pairs $(\boldsymbol{x}, \boldsymbol{y})$, we optimize the parameters $\boldsymbol{\theta}$ of a CNN denoiser $\mathcal{F}$, by minimizing for a loss function $\mathcal{L}(\cdot)$,

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \mathcal{L}(\boldsymbol{x}, \mathcal{F}(\boldsymbol{y};\boldsymbol{\theta})). \tag{1}$$

During transfer learning, networks parameters are optimized twice: initially with a pre-training dataset $\mathcal{S}$ and afterwords with a finetuning dataset $\mathcal{T}$. Given two different image noise datasets $\mathcal{S}$ and $\mathcal{T}$, collected using different cameras with noise distributions η_s and η_t, we define the denoising transfer learning framework with four key variables:

(i) the size of the pre-training dataset $\mathcal{S}$;

(ii) the size of the finetuning dataset $\mathcal{T}$;

(iii) the CNN parameter subset $\boldsymbol{\theta_t}$ updated for target dataset $\mathcal{T}$;

(iv) the similarity distance between the noise distributions of $\mathcal{S}$ and $\mathcal{T}$, denoted as $d(\eta_s, \eta_t)$.

To assess these variables we define which set of parameters we investigate, and how we compute the similarity distance between two different noise distributions. Lastly, we outline how we generate realistic synthetic-noise to simulate different cameras in our experiments.

3.1. Updating CNN parameters

The U-Net architecture has an encoder-decoder structure with a contracting $\mathcal{F}_E$ and expanding $\mathcal{F}_D$ part. The encoder part takes a noisy observation $\boldsymbol{y}$ to produce the inputs for the decoder, as

$$\mathcal{F}_{UNet}(\boldsymbol{y};\boldsymbol{\theta}) = \mathcal{F}_D(\mathcal{F}_E(\boldsymbol{y};\boldsymbol{\theta_E});\boldsymbol{\theta_D}), \tag{2}$$

where $\boldsymbol{\theta_E}$ and $\boldsymbol{\theta_D}$ are the encoder and decoder parameter subsets of the full paramater set $\boldsymbol{\theta}$.

We investigate the impact of restricted parameter updates during transfer learning, to prevent overfitting. Specifically we updated either the encoder $\boldsymbol{\theta_E}$ or the decoder $\boldsymbol{\theta_D}$.

3.2. Computing noise similarity distances

To evaluate the impact of noise similarity during transfer learning, we propose a methodology for retrieving the noise probability distributions, over which we can quantify similarity distances. We estimate a noise distribution using its probability mass function (PMF).

Consider an image signal with intensity within the range $[0., 1.]$. Let i be the intensity of a signal observation, then y_i is its noisy value and x_i is the ground truth value. The relation between y_i and x_i is given by

$$y_i = x_i + \eta_i, \tag{3}$$

where η_i is the noise of the signal. The signal noise η_i follows a probability distribution. Therefore, we can view y_i as an outcome of a random variable $\boldsymbol{Y}$, which describes the noise distribution.

To estimate the PMF of $\boldsymbol{Y}$ we use a frequency histogram $Hist(\boldsymbol{Y})$. We sample x_i uniformly from $[0., 1.]$, avoiding bias towards specific signal intensities. Assuming we know η_i, we compute y_i using Eq. (3). Finally, the number of bins of $Hist(\boldsymbol{Y})$ is set to 256, aligned with signal quantization which typically uses 8 bits to represent signal intensities. The resulting $Hist(\boldsymbol{Y})$ is a nominal type histogram [8], where each level or bin is independent. The PMF P for the noise distribution is produced by dividing each level by the number of samples generated N, $P = Hist(\boldsymbol{Y})/N$.

Given two noise distributions η_s and η_t, we derive their respective PMFs P_s and P_t to compute the noise similarity distance. In Tab. 1 we show the similarity distance measures for three different camera-specific noise distributions. The relative similarity distances between the camera noise distributions are independent of the metric chosen. In our experiments we quantify similarity using variation of information (Appendix A).

Camera-noise	Similarity distance to Omni camera-noise		
	Euclidian ↓	Normalized VI ↓	Wasserstein ↓
Sony A	0.047	0.803	$5.19 \cdot 10^{-4}$
Sony B	**0.014**	**0.621**	$\mathbf{1.47 \cdot 10^{-4}}$

Table 1. Comparing similarity distances between two different Sony camera-specific noises and an Omni camera-specific noise using three distinct metrics: (i) Euclidean distance; (ii) Normalized Variation of Information (VI); (iii) Wasserstein distance. All metrics indicate that Sony B camera noise is closer to the Omni camera noise, indicating that relative similarity distance is independent of the metric chosen.

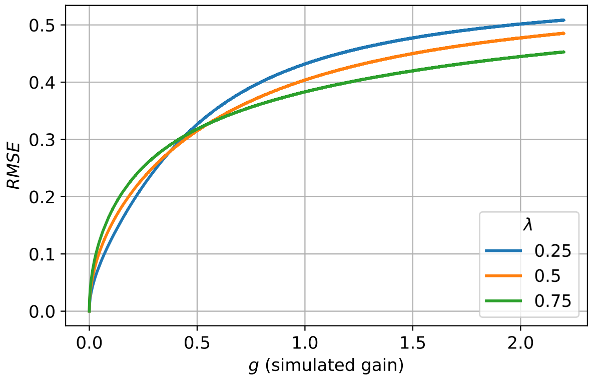


Figure 2. Poisson-Gaussian noise models with different λ show varying responses to changes in simulated gain g, as assessed through root mean squared error (RMSE) between sampled ground truth signal $\boldsymbol{x}$ and respective noisy signal $\boldsymbol{y}$. Thus, we can simulated distinct cameras by altering λ, and sample different noise levels varying simulated gain g.

3.3. Noise modelling

We generate realistic synthetic noise using the Poisson-Gaussian noise model [15], which combines two components: signal-dependent Poisson noise arising from photon sensing, and stationary signal-independent Gaussian noise. Extending Eq. (3), we define the relationship between a noisy signal value y_i and its respective ground truth value x_i using the Poisson-Gaussian noise model, as

$$y_i = x_i + \eta_p(x_i) + \eta_g, \tag{4}$$

where η_p is the Poisson noise component and η_g is the Gaussian noise component.

The two noise components are mutually independent, and their distributions can be characterized as follows,

$$(x_i + \eta_p(x_i))/a \sim \mathcal{P}(x_i/a), \quad \eta_g \sim \mathcal{N}(0, b), \tag{5}$$

where $a > 0$ and $b \geq 0$ are the parameters of the Poisson $\mathcal{P}$ and Gaussian $\mathcal{N}$ distributions. Parameter a is related to quantum efficiency, and can be interpreted as the signal value of a single detected photon. Parameter b represents the variance of the stationary noise.

Modelling different camera-specific noise. We control noise level of camera-specific noise models using a *gain* parameter. Higher levels of gain amplify signal values, increasing noise. Each camera-specific noise model responds differently to gain changes. To represent this relationship within a Poisson-Gaussian noise model, we introduce parameter $g > 0$, to simulate gain. We define the Poisson-Gaussian parameters a and b as a decomposition of g, characterized by a parameter $\lambda \in [0., 1.]$, as

$$a = \lambda g, \quad b = (1 - \lambda)g. \tag{6}$$

Variations in λ result in different responses of the Poisson-Gaussian noise model to changes in g, as depicted in Fig. 2. Altering λ is, therefore, akin to simulating different cameras. Notably, for $\lambda = (0.25, 0.5, 0.75)$, across the simulated gain g range $(0., 2.14]$, $\lambda = 0.25$ exhibits the most noise, while $\lambda = 0.75$ the least noise in terms of average variance (see Fig. 2).

Estimating real noise distribution. To compute the noise similarity distance we assume the noise distribution η is known, which is not true for images with real-noise. We estimate η for a real-noise image dataset using the Poisson-Gaussian noise model [3,4,15] (see details in Appendix B).

4. Experiments and Results

To investigate transfer learning in the context of denoising, we conduct our experiments in three distinct environments: **Toy-Set** (fully controlled camera noise distributions and datasets), **Color Natural-Images** (fully controlled camera noise distributions and real images), and **Camera-Specific Noise Models** (real camera-specific noise distributions and real images). Finally we extend to **Real-Noise Images** with unknown camera noise model to show applicability of noise distribution similarity.

Each experiment involves pre-training a network on images corrupted by a specific noise distribution and then finetuning the network on a different noise distribution. We conduct **Toy-Set** experiments 10 times with distinct network initialization, while experiments in the other environments are repeated 5 times. Experimental results are grouped by the investigated transfer learning variable.

Implementation details. We train the U-Net models using a combination of Charbonnier [10] ($\mathcal{L}_{Char}$) and SSIM [37] ($\mathcal{L}_{SSIM}$) loss,

$$\mathcal{L} = \mathcal{L}_{Char} + \mathcal{L}_{SSIM}. \tag{7}$$

The ADAM algorithm is used for optimization with mini batch size of 16. The models are trained for 1000 epochs, with initial learning rate 10^{-2}, which is then reduced by factor of 10^{-1} after every 400 epochs. Finetuning the models start with learning rate 10^{-3}. We use early stopping [34] selecting the model's weights at the epoch that achieved the highest validation set performance. Training images are cropped into random patches of size 256×256.

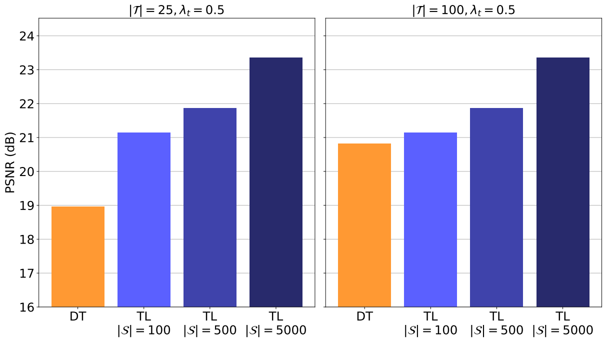


Figure 3. **Toy-Set.** Dataset size results for finetuning set sizes $|\mathcal{T}| = (25, 100)$ and $\lambda_t = 0.5$. Increasing the pre-training dataset size $|\mathcal{S}|$ improves performance. Transfer learning (TL) performance remains unaffected by changes in finetuning dataset size within the toy dataset. Yet, direct training (DT) exhibits significant improvement with more available data, as an increase of 75 images for training leads to a performance increase of ~ 1.5 dB.

4.1. Toy-Set

Experimental set-up. We create a toy dataset from FashionMnist [39]. Each image consists of 5 random fashion elements positioned at random non-overlapping locations. The images have a gray gradient background with pixel intensity in [0.,0.3]. The images have size of 128×128 pixels, therefore in the **Toy-Set**, we do not use random cropping. In addition, the models are trained for 100 epochs.

We simulate 3 different camera noise distributions using Poisson-Gaussian noise with $\lambda = (0.25, 0.5, 0.75)$. For each λ, we uniformly sample simulated gain g from the range $(0., 2.14]$, relative to pixel intensity $[0., 1.]$. We conduct experiments for each combination of λ_s and λ_t.

We vary the pre-training dataset sizes as $|\mathcal{S}| = (100, 500, 5000)$ and the finetuning dataset sizes as $|\mathcal{T}| = (25, 100)$. To examine the impact of dataset size, each experiment is executed for every pair of $|\mathcal{S}|$ and $|\mathcal{T}|$.

The final networks are evaluated against 1000 test images corrupted with the finetuning λ_t noise distribution.

Exp 1: Effect of dataset size. What is the effect of varying the size of the pre-training and finetuning dataset on transfer learning?

Results for $\lambda_t = 0.5$ are presented in Fig. 3. Transfer learning consistently outperforms training directly on the finetuning dataset. Increasing the pre-training set size improves transfer learning performance. Elevating the target set size from 25 to 100 instances, leads to a direct training performance increase of ~ 1.5 dB, but does not affect transfer learning performance. We conclude that having access to an extensive pre-training dataset is fundamental for denoising transfer learning, regardless of the availability of data during finetuning. Results for $\lambda_t = (0.25, 0.75)$ are

Parameter Set	Finetune λ_t (PSNR ↑ / SSIM ↑)		
	$\lambda_t = 0.25$	$\lambda_t = 0.5$	$\lambda_t = 0.75$
full	21.92 / 0.8565	22.12 / 0.8612	**22.75 / 0.8705**
encoder	21.83 / 0.8545	22.14 / 0.8587	22.59 / 0.8661
decoder	**21.93 / 0.8568**	**22.26 / 0.8632**	22.64 / 0.8700

Table 2. **Toy-Set.** Results for different sets of finetuned parameters evaluated on finetuning set size 100. Optimal performance is achieved by updating either the decoder or the all the parameters.

provided in Appendix C.

Exp 2: Updating the model's parameter. Does restricting the parameter set updated during transfer learning outperform updating all parameters by avoiding overfitting? We compare updating the encoder, decoder and the full parameter set of U-Net.

Updating only the decoder generally outperforms the encoder or full set updates. The difference between the encoder and decoder results is evident (Appendix D), with the decoder consistently outperforming the encoder for all metrics and finetuning distributions, as seen in Tab. 2. However, updating the entire parameter remains competitive. Finetuning the decoder achieves best results for $\lambda_t = (0.25, 0.5)$, but updating the full network achieves slightly better performance for $\lambda_t = 0.75$, by ~ 0.09 dB.

Notably restricting the finetuned parameter set is more efficient than adapting all parameters, as the networks converges faster to optimal performance (Appendix E).

Exp 3: Effect of noise distribution similarity. Can pretraining with a similar noise distribution improve performance on the finetuning distribution? We investigate similarity distances between the different simulated cameras λ distribution using normalized variation of information [33].

Results for finetuning $\lambda_t = 0.75$ are presented in Fig. 4. Finetuning similar noise distributions outperforms using less similar distributions. For pre-training dataset size $|\mathcal{S}| = 500$, conducting pre-training using the most similar noise distribution leads to an improved performance of ~ 1 dB.

Results for $\lambda_t = (0.25, 0.5)$ are provided in Appendix F.

4.2. Color Natural-Images

Experimental set-up. We evaluate if our findings still apply when using realistic color datasets. We use BSD400 [25], a standard natural-images dataset, for training. We pre-train the models on the entire dataset and finetune using a subset of 100 and 25 images. Similar to previous set-up, we simulate three different cameras with $\lambda = (0.25, 0.5, 0.75)$ and simulated gain g uniformly sampled from the range $(0., 2.14]$. The final networks are evaluated against test sets BSD68 [25] and Set14 [40], corrupted with the finetuning noise distribution.

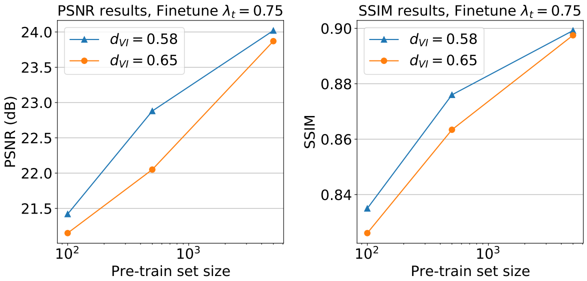


Figure 4. **Toy-Set.** Results for finetuning noise distribution $\lambda = 0.75$, highlighting similarity distance by variation of information d_{VI}. Optimal performance is attained through finetuning with similar noise distributions.

Parameter Set	BSD68, $\lvert\mathcal{T}\rvert = 25$ (PSNR ↑ / SSIM ↑)		
	$\lambda_t = 0.25$	$\lambda_t = 0.5$	$\lambda_t = 0.75$
full	20.77 / 0.5605	20.57 / 0.5697	21.05 / 0.5932
encoder	20.79 / 0.5591	20.50 / 0.5686	20.78 / 0.5897
decoder	**20.84 / 0.5607**	**20.62 / 0.5702**	**21.17 / 0.5947**

Table 3. **Color Natural-Images.** Results for different sets of updated parameters evaluated on finetuning set size $|\mathcal{T}| = 25$. Updating the decoder achieves optimal performance.

Exp 1: Updating the model's parameter. Does finetuning only the decoder outperform updating the entire parameter set when using realistic images?

Adapting only the decoder consistently outperforms updating the full parameter set when the finetuning set size is limited. Tab. 3 presents the results for test set BSD68 using finetune set size $|\mathcal{T}| = 25$. The table shows that the decoder outperforms adapting the encoder or the full network for all target distributions λ_t. However, as the finetune set size increases, we find that updating the full parameter set exhibits improved performance.

In conclusion, adapting the entire parameter set of U-Net leads to overfitting when the finetuning dataset size is small. In such cases, updating only the decoder proves to be the optimal strategy.

Exp 2: Effect of noise distribution similarity. Does pre-training on noise distributions similar to the finetuning noise improve performance with real images? By using standard natural-images dataset BSD400, we extend the results applicability to diverse image signals.

Our findings indicate that noise similarity positively impacts performance. In Fig. 5 we highlight the results for finetuning noise distributions $\lambda_t = 0.25$, which represents the most noisy simulated camera, and $\lambda_t = 0.75$, which represents the least noisy simulated camera. For test dataset BSD68, finetuning with the most similar noise distribution improves performance by ~ 0.2 dB for $\lambda_t = 0.25$ and by ~ 0.6 dB for $\lambda_t = 0.75$. The results hold consistent

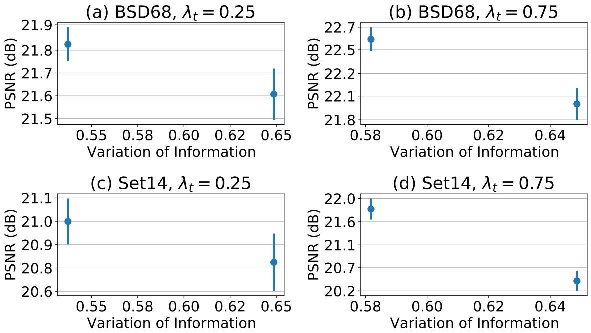


Figure 5. **Color Natural-Images.** Transfer learning performance results as mean and standard deviation, for finetuning $\lambda_t = (0.25, 0.75)$, evaluated using the BSD68 and Set14 test sets. Similarity distances are computed with normalized variation of information. Here, $\lambda_t = 0.25$ represents the *most noisy* simulated camera, while $\lambda_t = 0.75$ corresponds to the *less noisy* simulated camera. The results consistently demonstrate that noise similarity improves performance in both directions, finetuning from less noisy to more noisy distributions, or vice versa

across test dataset Set14, where noise similarity leads to an enhancement of $\sim$ 0.2 and $\sim$ 1.5 dB for $\lambda_t = 0.25$ and $\lambda_t = 0.75$ respectively.

4.3. Camera-Specific Noise Models

Experimental set-up. To validate our findings, we employ real camera-specific noise models from two distinct Sony cameras (referred to as Sony A and Sony B) as well as an Omnivision (Omni) camera. Unlike the simulated Poisson-Gaussian camera model where the gain is capped at a maximum of 2.14 due to noise variance plateauing, real camera-specific noise models exhibit a much wider gain range. For each camera-specific noise model we uniformly sample gain from the range $[0, 1024]$, relative to pixel intensity $[0., 1.]$). We use the same datasets set-up as in the **Color Natural-Images** environment.

Exp 1: Updating the model's parameter. How is performance affected by restricting the parameter set updated during transfer learning when using real camera-specific noise distributions? We pre-train U-Net on one camera's noise and finetune across the other distributions.

Finetuning only the decoder parameters generally outperforms updating the full parameter set, when the finetune set size is limited. Results for finetuning camera Omni are presented in Fig. 6, which illustrates that updating the full network can lead to overfitting and lower performance compared to restricting the adapted parameter set. Additional results for finetuning camera Sony A and Sony B are reported in Appendix G.

Exp 2: Effect of noise distribution similarity. Can noise similarity enhance transfer learning using real camera-specific noise distributions? We evaluate whether pre-training on a similar target camera noise distribution leads to improved results compared to less similar distributions. Results for finetuning camera Sony A are presented in Tab. 4. Our findings indicate that pre-training with similar noise enhances performance, even outperforming the use of combined noise models.

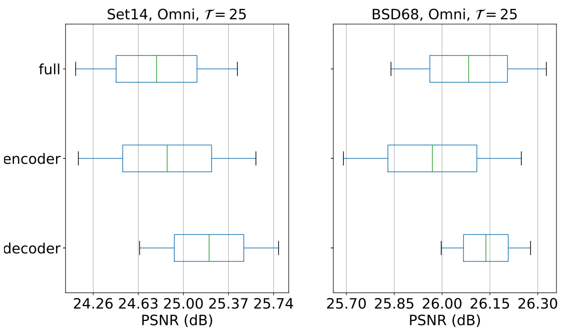


Figure 6. **Camera-Specific Noise Models.** Results for target camera Omni and finetune set size 25. Updating only the decoder prevents overfitting and achieves the best performance.

Pre-train Camera	$d_{VI} \downarrow$	Finetuning Sony A (PSNR ↑ / SSIM ↑)	
		Set14	BSD68
Omni, Sony B	-	24.59 / 0.7739	27.54 / 0.8162
Omni	0.80	23.81 / 0.7721	26.85 / 0.8132
Sony B	**0.76**	**27.18 / 0.7984**	**27.85 / 0.8220**

Table 4. **Camera-Specific Noise Models.** Results for finetuning camera Sony A. Models pre-trained with Sony B, the most similar camera noise, achieve the highest performance, outperforming models pre-trained with Omni camera noise or on a set of images featuring both noise distributions.

In Appendix H we present the results for cameras Sony B and Omni, while in Appendix I we extend our findings to diverse finetuning data.

4.4. Real-Noise Images

Experimental set-up. We investigate applicability of noise similarity in the context of real-noise images with unknown camera noise distribution. We pre-train the U-Net models using the BSD400 [25] dataset. As finetuning data, we employ the real-noise datasets SSID-Small [1] and RENOIR [3], specifically utilizing images from the LG G4 camera camera in SSID-Small and images from the Xiaomi Mi3 camera in the RENOIR dataset. SSID-Small contains 16 clean-noisy image pairs from 8 distinct image scenes captured using the LG G4 camera. These are divided into 6 image pairs for training, 4 for validation, and 6 for testing. Similarly, RENOIR contains 79 clean-noisy image pairs from 40 image scenes captured using the Xiaomi Mi3

Technique	Pre-train Noise	Finetuning LG G4			Finetuning Xiaomi Mi3		
		$d_{VI}\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	$d_{VI}\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
Direct training	-	-	25.51	0.8894	-	26.24	0.7862
Transfer learning	$\lambda_s = (0.25, 0.5, 0.75)$	-	31.19	0.9178	-	30.69	0.8175
Transfer learning	$\lambda_s = 0.25$	0.77	31.16	0.9113	0.82	28.75	0.8055
Transfer learning	$\lambda_s = 0.75$	0.62	31.59	0.9214	0.67	30.19	0.8137
Transfer learning	$\lambda_s = 0.5$	**0.39**	**35.03**	**0.9448**	**0.64**	**31.16**	**0.8180**

Table 5. **Real Noise Images.** Average PSNR and SSIM results for test sets LG G4 and Xiaomi Mi3. Transfer learning enhances performance in comparison to direct training on the target datasets. Networks trained on noise distributions similar to the target outperform those trained on less similar or varied noise sources. Notably, greater noise similarity correlates with higher performance gains.

camera. This dataset is partitioned into 47 image pairs for training, 12 for validation, and 20 for testing.

LG G4 images have 2988×5312 pixels, while Xiaomi Mi3 images have 3000×3000 pixels. BSD400 images have generally 481×321 pixels. A single image from the real-noise datasets has more than 50 times the amount of data present in a BSD400 image. Therefore, we update the entire parameter set during transfer learning since, in this context, we cannot apply the numerical boundaries we found in the previous experiments for preferring finetuning the decoder over the whole parameter set.

Since we estimate the noise distribution of the real-noise images using the Poisson-Gaussian noise model, we use the Poisson-Gaussian simulated cameras for U-Net pre-training, in order to fairly compare noise similarity (Appendix K). We use three different simulated camera distributions with $\lambda = (0.25, 0.5, 0.75)$ and simulated gain g uniformly sampled from the range $(0., 2.14]$.

Exp 1: Effect of noise distribution similarity. Does noise similarity enhance transfer learning using real-noise images? The results for finetuning cameras LG G4 and Xiaomi Mi3 are presented in Tab. 5. For the LG G4, models pre-trained on the most similar noise distribution show an improvement of approximately $\sim$ 10 dB compared to models directly trained on the target images, and around $\sim$ 3.5 dB compared to models pre-trained on the second most similar noise. The models performance follows an order determined by similarity distance. Interestingly, the models trained on the most similar noise distribution outperform the models trained on pre-training data containing all available noise distributions.

These findings indicate that noise similarity significantly improves real-noise transfer learning performance. We expand the robustness of these results by considering confounding variables (Appendix J) and evaluating the significance of real-noise distribution estimates (Appendix K). Furthermore, our insights extend beyond the encoder-decoder framework (Appendix L), establishing the broad applicability of noise similarity in transfer learning.

5. Limitations and Conclusion

Our findings regarding which U-Net parameter set to update during transfer learning depend on the scale of the images used during experimentation. However, real-world image data is typically significantly larger, limiting direct applicability of the dataset sizes bounds where overfitting occurs. Nevertheless, contemporary CNN denoisers employ more parameters than U-Net, leading to overfitting challenges even with large real-world data. Therefore our emphasis on updating the expanding part of the encoder-decoder network structure remains relevant.

Real world noise originates from different sources, besides the sensor size or type, like JPEG compression. These factors cannot be well modeled solely by the employed Poisson-Gaussian noise distribution. Our similarity findings rely on the capacity of this distribution to model real image noise, and thus, they are constrained by it. Furthermore, our comparison of noise similarity assumes image noise to be channel and spatially independent. This may not hold in the real-noise images, in which case a more complex method needs to be employed to compute the PMF of a noise distribution.

To conclude, we presents a comprehensive analysis encompassing four key components of transfer learning in the denoising domain. We highlight the significance of pre-training and finetuning dataset size. We demonstrate how a limited finetuning set size can lead to overfitting during parameters update. We show that finetuning the decoder of a U-Net architecture produces better denoising results and convergence rates, compared to updating the encoder. Moreover, we introduce the concept of noise distribution similarity in denoising transfer learning. Our empirical results demonstrate noise similarity has a significant impact on performance improvement, effectively reducing domain discrepancy between the pre-train noise domain and the finetune noise domain.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 1, 2, 7

[2] Mina Amiri, Rupert Brooks, and Hassan Rivaz. Fine tuning u-net for ultrasound image segmentation: which layers?, 2020. 3

[3] Josue Anaya and Adrian Barbu. Renoir – a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018. 1, 2, 3, 5, 7, 11, 14, 15

[4] Nicolas Bahler, Majed El Helou, Etienne Objois, Kaan Okumus, and Sabine Susstrunk. PoGaIN: Poisson-gaussian image noise modeling from paired samples. *IEEE Signal Processing Letters*, 29:2602–2606, 2022. 3, 5

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

[6] HC. Burger, CJ. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? pages 2392 – 2399, June 2012. 1

[7] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. 2007. 3

[8] Sung-Hyuk Cha and Sargur N. Srihari. On measuring the distance between histograms. *Pattern Recognit.*, 35:1355–1370, 2002. 4

[9] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising, 2020. 2

[10] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proceedings of 1st International Conference on Image Processing*, 2:168–172 vol.2, 1994. 5

[11] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. 2018. 1

[12] Dorothy Cheng and Edmund Y. Lam. Transfer learning u-net deep learning for lung ultrasound segmentation, 2021. 3

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 2

[14] Chi-Mao Fan, Tsung-Jung Liu, and Kuan-Hsien Liu. SUNet: Swin transformer UNet for image denoising. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, may 2022. 2

[15] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. 2, 3, 4, 5, 10, 11

[16] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[17] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[19] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015. 13, 14

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 2

[21] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision, 2021. 2

[22] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3

[23] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. 3

[24] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. 3

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik. 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*, volume 2, pages 416–423 vol.2, 2001. 6, 7, 13, 14

[26] Sreyas Mohan, Zahra Kadkhodaie, Eero P. Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks, 2020. 2

[27] Sreyas Mohan, Joshua L. Vincent, Ramon Manzorro, Peter A. Crozier, Eero P. Simoncelli, and Carlos Fernandez-Granda. Adaptive denoising via gaintuning, 2021. 2, 3

[28] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs, 2017. 2

[29] K. Ram Prabhakar, Vishal Vinod, Nihar Ranjan Sahoo, and R. Venkatesh Babu. Few-shot domain adaptation for low light raw image enhancement. 2023. 1

[30] Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. On wasserstein two sample testing and related families of non-parametric tests, 2015. 3

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[32] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. 3

[33] Kihyuk Sohn, Wenling Shang, and Honglak Lee. Improved multimodal deep learning with variation of information. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 3, 6, 10

[34] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 3, 5

[35] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. *CoRR*, abs/1711.10925, 2017. 2

[36] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Lidia: Lightweight learned image denoising with instance adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2, 3

[37] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[38] K. Wei, Y. Fu, J. Yang, and H. Huang. A physics-based noise formation model for extreme low-light raw denoising. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2755–2764, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 1

[39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 5

[40] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In Jean-Daniel Boissonnat, Patrick Chenin, Albert Cohen, Christian Gout, Tom Lyche, Marie-Laurence Mazure, and Larry Schumaker, editors, *Curves and Surfaces*, pages 711–730, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 6

[41] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1, 2, 14, 15

[42] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 1

[43] Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. A poisson-gaussian denoising dataset with real fluorescence microscopy images, 2019. 3, 10

A. Variation of Information

Variation of Information (VI) is an information theoretic metric which quantifies the information distance between two random variables [33]. It is closely related to mutual information, with the difference that VI is a true metric. We use VI to measure the similarity distance between two probability mass functions (PMFs) representing two distinct noise distributions. VI can be computed by the sum of the conditional entropy of the distributions, each ”measuring” the distance between one distribution to the other.

The VI between two PMFs P and Q is defined as:

$$VI(P, Q) = H(P|Q) + H(Q|P), \tag{8}$$

where $H(P|Q)$ is the conditional entropy of distribution P given Q and $H(Q|P)$ is the conditional entropy of distribution Q given P. The conditional entropy of a distribution is defined as follows,

$$H(P|Q) = -\sum_{y_p, y_q} P(y_p, y_q)\log(\frac{P(y_p, y_q)}{Q(y_q)}), \tag{9}$$

where y_p, y_q range over the signal intensities for which the PMFs are defined, $P(y_p, y_q)$ is the joint probability of signal intensities y_p and y_q according to distribution P, and $Q(y_q)$ is the probability of signal intensity y_q according to distribution Q.

In our experiments we use normalized VI. We normalize VI by dividing the metric by the joint entropy of the two distributions. The normalized value will then be within the interval $[0., 1.]$, simplifying comparisons.

B. Estimating Poisson-Gaussian noise parameters from real-noise images

Based on the properties of the Poisson and Gaussian distributions [15, 43] the mean μ and variance σ^2 of noisy signal y_i with ground truth signal x_i, can be written as

$$\mu(y_i) = x_i, \quad \sigma^2(y_i) = ax_i + b. \tag{10}$$

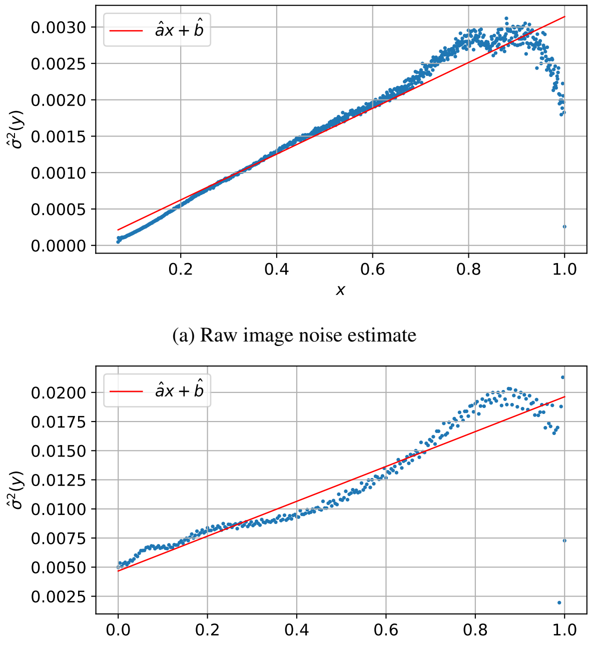
Consider a real clean-noisy image pair. We define $\mathcal{Y}_i$ as the set containing all noisy signals y_i for a specific clean signal x_i with signal intensity i. Since by Eq. (10) the mean of $\mathcal{Y}_i$ is x_i, we empirically compute the variance $\hat{\sigma}^2$ of y_i,

$$\hat{\sigma}^2(y_i) = \frac{1}{|\mathcal{Y}_i|}\sum_{y_i \in \mathcal{Y}_i}(y_i - x_i)^2. \tag{11}$$

Finally since $\sigma^2(y_i) = ax_i + b$ (Eq. (10)), we can estimate parameters a, b of the real-noise image using the least squares method,

$$\hat{a}, \hat{b} = \arg\min_{a,b}\sum_i(\hat{\sigma}^2(y_i) - ax_i - b)^2. \tag{12}$$

In Fig. 7 we show the results of the estimation method for an image pair from the RENOIR [3] dataset.



(a) Raw image noise estimate

(b) RGB image noise estimate

Figure 7. Noise variance as a function of ground truth signal intensity, estimated from a RENOIR test image. We show the noise estimation using (a) the RAW version of the image pair and (b) the RGB version of the image pair. Non-linearity near the boundaries, is likely the result of clipping [15].

B.1. Is it appropriate to estimate Poisson-Gaussian real-noise from RGB images?

Raw images represent unprocessed sensor data, while RGB images result from sensor data undergoing several processing steps, including color correction. In this study, we estimate Poisson-Gaussian parameters from RGB images, prompting the question of its appropriateness.

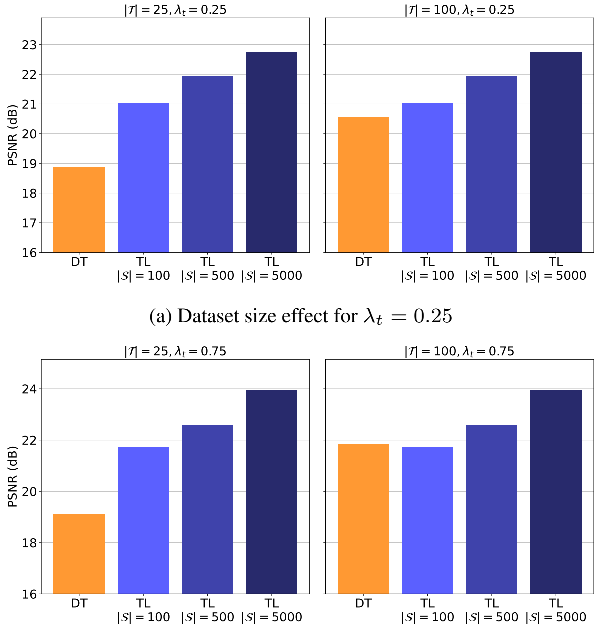
Raw images directly reveal noise characteristics in relation to sensor data, but raw image datasets are scarce. Furthermore, attempting to reconstruct raw data from RGB images only produces an approximation, given complex or unknown image processing pipelines.

Therefore, estimating real-noise from RGB images offers practical advantages, and we show is sufficient to identify optimal pre-training noise distributions for finetuning with real-noise images.

C. Toy-Set: dataset size effect results for $\lambda_t = (0.25, 0.75)$

In Fig. 8 we present the dataset size effects results for $\lambda_t = 0.25$ and $\lambda_t = 0.75$. As outlined in Sec. 4.1 increasing the finetuning set size $|\mathcal{T}|$ leads to direct training increasing performance by 1 to 2 dB. However, transfer learning performance remains unaffected. Conversely increasing the pre-training dataset size $|\mathcal{S}|$ significantly impacts transfer learning performance, with ~ 1 dB improvement from $|\mathcal{S}|$ 100 to 500.

Notably, when $\lambda_t = 0.75$ and $|\mathcal{T}| = 100$, direct training slightly outperforms transfer learning with only 100 pre-training instances. Nonetheless, the difference is marginal, as the data available for both direct training and transfer learning is quite comparable. In all other cases, transfer learning outperforms direct training, validating the potential of the method.



(a) Dataset size effect for $\lambda_t = 0.25$

(b) Dataset size effect for $\lambda_t = 0.75$

Figure 8. **Toy-Set.** Dataset size results with finetuning set sizes $|\mathcal{T}| = (25, 100)$, for (a) $\lambda_t = 0.25$ and (b) $\lambda_t = 0.75$. Increasing pre-train set size $|\mathcal{S}|$ significantly enhances transfer learning (TL) performance. While extending the finetuning set size $|\mathcal{T}|$ impacts direct training (DT) performance, it has minimal effect on transfer learning performance.

D. Updating individual U-Net layers

We investigate our results indicating that finetuning the decoder outperforms finetuning the encoder, by updating each singular layer of the U-Net architecture while fixing

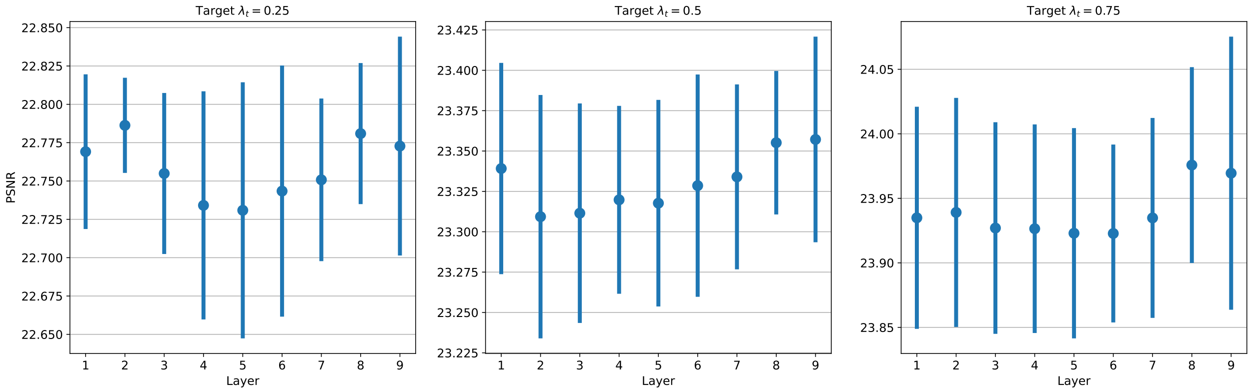


Figure 9. **Toy-Set.** Results showing mean and standard deviation of performance obtained by updating a single layer. Layers are numbered as they appear in the U-Net architecture following an input signal. The last layers typically perform better than the first.

the remaining layers.

We perform our analysis on the toy dataset using pre-training set size 5000 and different finetuning set sizes (100, 50, 25). We keep all other experimental set-up parameters as in Sec. 4.1.

The results are presented in Fig. 9. We find that updating the final layers generates better results on average, providing insight into the decoder's superior performance in the denoising domain.

E. Convergence rate of different parameter sets

We analyse the time required for finetuning, while updating distinct parameter sets. We measure time in terms of epochs taken to converge to the best model relative to the validation dataset. We gather the results using the **Toy-Set** experimental set-up (Sec. 4.1).

The results are presented in Fig. 10. While there are no notable differences in convergence speed with finetuning dataset size of 100, as the finetuning dataset size decreases, updating only the decoder becomes the most efficient configuration for transfer learning. When the finetuning dataset contains 25 instances, the difference in epoch needed for convergence between updating the whole network and the decoder alone is on average ~ 20.

F. Toy-Set: noise similarity effect results for $\lambda_t = (0.25, 0.5)$

In Fig. 11 we present the noise similarity effects results for $\lambda_t = 0.25$ and $\lambda_t = 0.5$. The results follow the trend discussed in Sec. 4.1, wherein pre-training with a noise distribution similar to the finetuning distribution leads to performance enhancement.

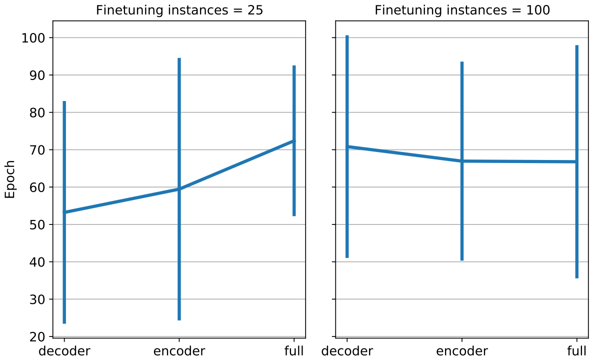
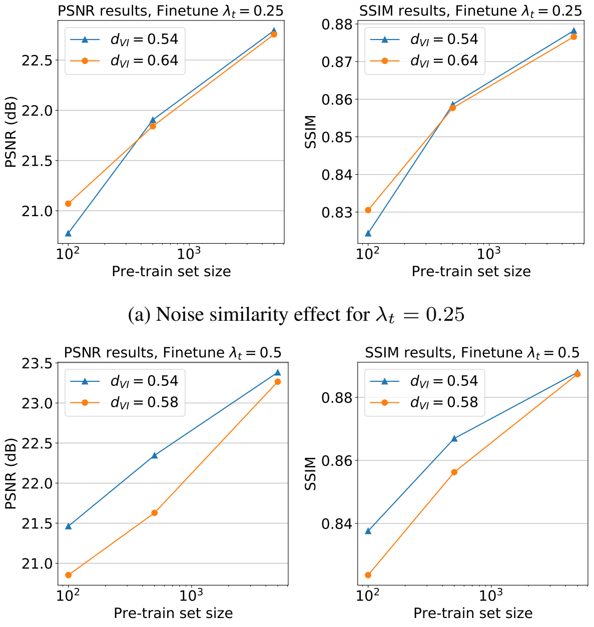


Figure 10. **Toy-Set.** Epoch count taken by each parameter set to reach the best model relative to the validation dataset. With smaller finetuning dataset sizes, the decoder converges faster compared to the other configurations.

Notably, instances for $\lambda_t = 0.25$ do not exhibit significant performance difference. Furthermore, at pre-training set size $|\mathcal{S}| = 100$, utilizing less similar noise actually produces improved results. However, it's important to emphasize that this behavior is isolated to a single data point.

G. Camera-Specific Noise Models: parameters set effect results for finetuning cameras Sony A and Sony B.

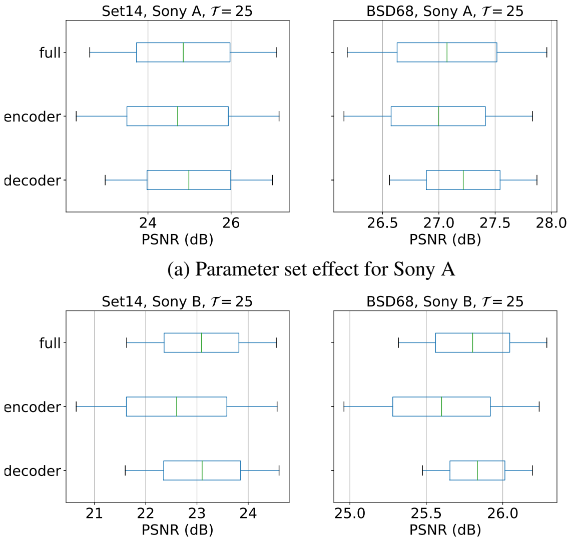
In Fig. 12 we present the results relative to parameters set update for finetuning cameras Sony A and Sony B. As discussed in Sec. 4.3 adapting only the decoder generally is more stable and performs better, compared to the encoder or the whole network.



(a) Noise similarity effect for $\lambda_t = 0.25$

(b) Noise similarity effect for $\lambda_t = 0.5$

Figure 11. **Toy-Set.** Noise similarity results for (a) $\lambda_t = 0.25$ and (b) $\lambda_t = 0.5$. Generally pre-training on noise similar to the finetuning distribution improves performance.



(a) Parameter set effect for Sony A

(b) Parameter set effect for Sony B

Figure 12. **Camera-Specific Noise Models.** Updating different parameter's set of U-Net for cameras (a) Sony A and (b) Sony B. The decoder generally outperforms the other configurations.

H. Camera-Specific Noise Models: noise similarity effect results for finetuning cameras Sony B and Omni.

In Tab. 6 and Tab. 7 we present, respectively, the noise similarity results for finetuning cameras Sony B and Omni. Noise similarity properties hold for both cameras, since the most similar noise distribution achieves the highest performance across all metrics. These findings indicate noise similarity can be applied beyond simple Poisson-Gaussian camera noise models.

Pre-train Camera	d_{VI} ↓	Finetuning Sony B (PSNR ↑ / SSIM ↑)	
		Set14	BSD68
Sony A	0.76	22.05 / 0.7355	25.69 / 0.7751
Omni	**0.62**	**24.88 / 0.7485**	**26.37 / 0.7778**

Table 6. **Camera-Specific Noise Models.** Results for finetuning camera Sony B. Models pre-trained with Omni, the most similar camera noise, achieve the highest performance.

Pre-train Camera	d_{VI} ↓	Finetuning Omni (PSNR ↑ / SSIM ↑)	
		Set14	BSD68
Sony A	0.80	24.45 / 0.7339	25.69 / 0.7778
Sony B	**0.62**	**25.89 / 0.7551**	**26.85 / 0.8132**

Table 7. **Camera-Specific Noise Models.** Results for finetuning camera Omni. Models pre-trained with Sony B, the most similar camera noise, achieve the highest performance.

I. Noise similarity with different finetuning image content

Does noise similarity continue to enhance performance when the finetuning images have different characteristics from the pre-traininng images? We investigate the effect of noise similarity using the Urban100 dataset [19], which is composed by urban scene images with high-self similarity. The dataset introduces a content shift compared to the natural-images dataset BSD400 [25], which we use as our pre-training dataset. We hold out 40 images from Urban100 for testing. We keep all other experimental set-up parameters as in Sec. 4.3.

We present results for finetuning camera Sony A in Tab. 8. Networks pre-trained with Sony B camera noise, the most similar noise distribution to Sony B, achieve the highest performance, with improvement of ∼ 0.5 dB. The result indicate that even when there is a domain shift in image content between the pre-training and finetuning datasets, noise distribution similarity still enhances performance.

Pre-train Camera	$\mathcal{T}$ Data	$\mathcal{T}$ Camera	d_{VI} ↓	PSNR ↑	SSIM ↑
Omni, Sony B	Urban images	Sony A	-	27.09	0.8299
Omni	Urban images	Sony A	0.80	26.99	0.8288
Sony B	Urban images	Sony A	**0.76**	**27.56**	**0.8383**

Table 8. **Camera-Specific Noise Models.** Results for finetuning $\mathcal{T}$ camera Sony A using Urban100 [19] as finetuning set and BSD400 [25] as pre-training set. Results indicate that even with an image content domain shift noise distribution similarity enhances transfer learning performance.

J. Effect of pre-training signal range

In our experiment with real-noise images we pre-train our models with the BSD400 [25] dataset. Images from BSD400 have normal brightness conditions, while images with real-noise are typically taken in dark conditions. What is the effect of pre-training on a dataset with a wider signal range relative to the finetuning dataset?

We create a copy of the BSD400 dataset, by scaling the signal range to the one present in real-noise images from the Xiaomi Mi3 camera, effectively creating a *darker* version of BSD400. We pre-train U-Net on the normal and dark version of BSD400. We use the Poisson-Gaussian noise distribution estimate of the Xiaomi Mi3 image noise, which is defined at a specific noise level.

Average results for finetuning dataset Xiaomi Mi3, are presented in Tab. 9. Pre-training U-Net with BSD400 under normal brightness conditions outperforms pre-training using the down scaled signal version of the dataset. The results indicates that a wider signal range enhances transfer learning performance when finetuning with a dark dataset. The finding supports our strategy of using a dataset with normal brightness conditions as BSD400 in our experiments with real-noise transfer learning.

Pre-Train Data	Pre-Train Noise	Finetuning Xiaomi Mi3	
		PSNR ↑	SSIM ↑
Dark BSD400	Xiaomi Mi3 Estimate	27.36	0.7868
Normal BSD400	Xiaomi Mi3 Estimate	**28.93**	**0.7984**

Table 9. **Real-Noise Images.** Average results for the Xiaomi Mi3 test set by pre-training on the normal BSD400 dataset or its darker version (with signal range matching the Xiaomi Mi3 data). Pre-training on a brighter dataset enhances transfer learning performance when finetuning on darker images.

K. Pre-training on real-noise distribution estimate

Does pre-training using the real-noise Poisson-Gaussian distribution estimate outperform pre-training with the synthetic camera models? We compare pre-training using simulated Poisson-Gaussian camera noise model with pre-training using the Poisson-Gaussian distribution estimating the noise of the real-noise finetuning dataset.

We use $\lambda = (0.25, 0.5, 0.75)$ and sample simulated gain g uniformly from range $(0., 2.14]$. We estimate the noise distribution of Xiaomi Mi3 (camera from RENOIR [3]) as detailed in Appendix B. We pre-train U-Net on each noise distribution using the BSD400 [25] dataset. We then finetune the models on the Xiaomi Mi3 real-noise images.

Average results are presented in Tab. 10. Pre-training with the estimated real-noise distribution under-performs compared to pre-training using the simulated camera models. Since pre-training simulated camera noise is defined over a range of parameter g, the results indicate that pretraing on a wide range of noise levels has a significant positive effect on finetuning performance.

This finding highlights the importance of evaluating noise similarity. While several techniques are available to estimate the noise distribution parameters of real-noise images, they are only able to do so for a specific noise level. By evaluating noise similarity against simulated camera noises, we are able to select a pre-training noise distribution for which we can model noise at different levels.

Computing noise similarity between the estimated real-noise distribution and the available Poisson-Gaussian distributions, is akin to using the estimated a and b parameters to retrieve the λ parameter that best simulates the camera which captured the real-noise images.

Pre-Train Noise	g Interval	Finetuning Xiaomi Mi3	
		PSNR ↑	SSIM ↑
Xiaomi Mi3 Estimate	-	28.93	0.7984
$\lambda = 0.25$	$(0., 2.14]$	28.75	0.8055
$\lambda = 0.75$	$(0., 2.14]$	30.19	0.8137
$\boldsymbol{\lambda = 0.5}$	$(0., 2.14]$	**31.16**	**0.8180**

Table 10. **Real-Noise Images.** PSNR and SSIM results for Xiaomi Mi3 test images. Pre-training on the simulated camera models outperforms pre-training on the real-noise distribution estimate. We note that the simulated camera models are defined for wide noise levels, while the real-noise distribution estimate is defined for a single noise level. The results highlight the importance of noise similarity in denoising transfer learning, since it allows us to select a noise distribution similar to the finetuning noise, but for which we can simulate several noise levels.

L. Assessing real-noise similarity with DnCNN [41]

Does noise similarity enhance transfer learning performance when applied to network architectures beyond the traditional encoder-decoder structure? To address this, we replicate the experiments outlined in Sec. 4.4, focusing on images captured with the Xiaomi Mi3 camera from the RENOIR dataset [3]. However, we now employ DnCNN [41], a popular network architecture specifically designed for image denoising. DnCNN leverages a sequence of convolutional layers with residual connections and does not have an encoder-decoder structure.

Our findings, presented in Tab. 11, closely resemble those obtained using the U-Net architecture in Sec. 4.4. Pre-training with noise distributions similar to the target real-noise significantly enhances performance. Notably, pre-training with the most similar noise distribution outperforms pre-training with all available noises. Furthermore, transfer learning continues to demonstrate substantial performance gains over direct training on the Xiaomi Mi3 dataset.

In conclusion, our results highlight the adaptability of noise similarity across different CNNs architectures.

Technique	Pre-train Noise	Finetuning Xiaomi Mi3		
		d_{VI} ↓	PSNR ↑	SSIM ↑
Direct training	-	-	23.17	0.7581
Transfer learning	$\lambda_s = (0.25, 0.5, 0.75)$	-	29.48	0.8095
Transfer learning	$\lambda_s = 0.25$	0.77	29.26	0.8081
Transfer learning	$\lambda_s = 0.75$	0.62	29.56	0.8090
Transfer learning	$\lambda_s = 0.5$	**0.39**	**29.84**	**0.8104**

Table 11. **Real Noise Images.** PSNR and SSIM results for test set Xiaomi Mi3 using DnCNN architecture. Similarity noise correlates with higher performance gains. This indicates noise similarity effectiveness extends beyond encoder-decoder architectures.

3

Supplemental Materials

3.1. Image Noise

Digital images are the result of digital cameras capturing light patterns (photons) from scenes and converting them into digital forms. However, this process introduces noise. Signal noise is a well-studied phenomenon within the field of signal processing, as it degrades the quality and accuracy of the original signal.

This thesis centers on noise removal from digital images. Thus, it is important to establish a foundational understanding of image noise, its nature, sources, and how we measure it.

3.1.1. What is image noise?

Image noise is defined as a random variation of brightness or color information among an image's pixels, appearing as a grainy structure covering the image (see Figure 3.1). These fluctuations mask the original signal values and inevitably degrade image quality. The presence of image noise complicates the identification of important image features, which has a negative impact on several applications such as camera surveillance, medical imaging and object detection.

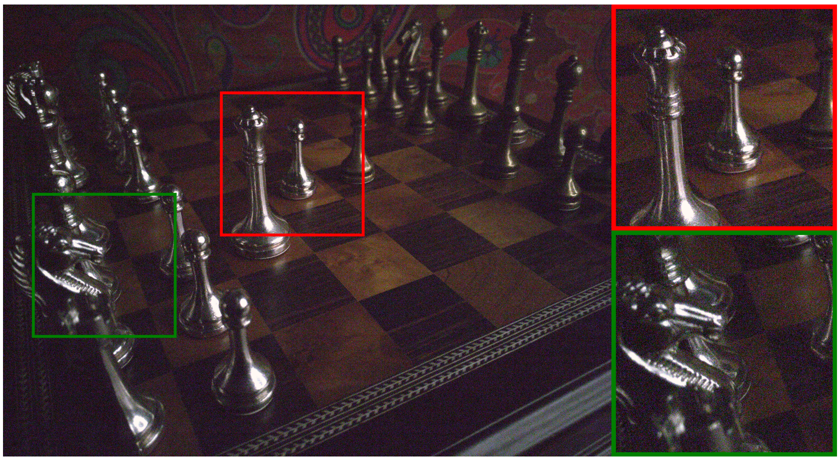


Figure 3.1: Noisy image captured with a Samsung Galaxy S6 Edge camera [1]. Noise appears as a grainy random structure covering the image.

Noise emerges from a range of sources, including low lighting, long exposure times, heat, and faults within the image sensor. Broadly speaking, noise can be divided into two main categories: signal-dependent noise and signal-independent noise. Signal-dependent noise, as the term implies, depends on the intensity of incoming photon signals. Conversely, signal-independent noise maintains a constant variance and is influenced by factors as image sensor dimensions or type. Within these two noise categories, shot noise and read noise are typically the most prevalent. We will focus on them, since they lay the foundation for understanding the nature of image noise.

Shot Noise

Digital cameras use an image sensor to capture images. The image sensor is composed by an array of pixels. Pixels are tiny light-sensitive units, which accumulate electric charge upon photon impact. This charge is then converted into digital information. Shot noise emerges from the inherent randomness of photons. Their arrival times at any pixel location is unpredictable, thus introducing variability into the measurements (see Figure 3.2).

The photon-induced shot noise is well fitted by the discrete Poisson distribution [28]. The Poisson distribution models the uncertainty of a given number of events occurring over a period of time. In this context, these events are the actual photons hitting the pixels at specific times. Hence, the Poisson distribution is well suited to model shot noise.

Read Noise

Read noise emerges as the camera converts accumulated pixel charge into voltage. Read noise is, therefore, a measurement error and depends on factors as the camera quality and its electronic circuits.

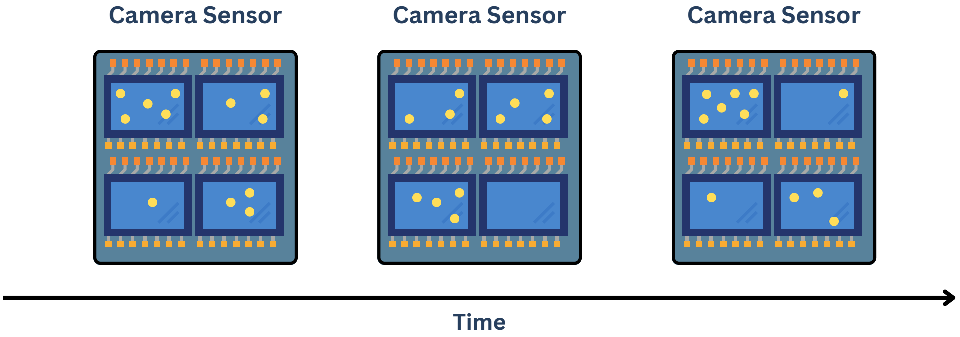


Figure 3.2: Illustrating shot noise. At different time units, the number of photons hitting the pixels from the same light source varies.

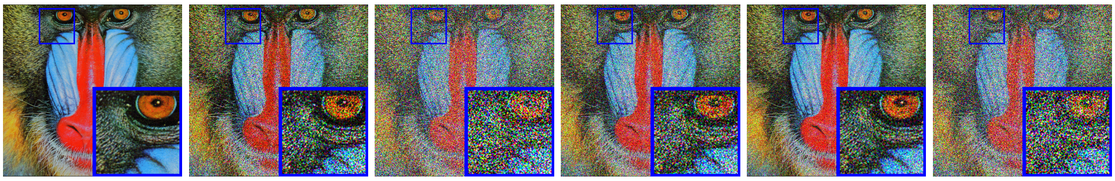
Importantly, read noise does not depend on the signal intensity, and thus it is signal-independent [3]. Modelled as a Gaussian distribution with a zero-mean, its standard deviation determines the noise level.

Combining Shot and Read Noise, the Poisson-Gaussian noise model.

The signal-dependent and the signal-independent nature of image noise are combined in the Poisson-Gaussian noise model distribution [8]. The model is composed by a Poisson component $\mathcal{P}$ which captures the signal-dependent part of the noise, α, and a Gaussian $\mathcal{N}$ component which captures the signal-independent part of the noise, β. The distribution is parameterized with factors a and b, outlined as:

$$\alpha \sim \mathcal{P}(x/a), \quad \beta \sim \mathcal{N}(0, b), \tag{3.1}$$

where x represents the clean signal value. a can be interpreted as the signal value of a single detected photon [28], while b is simply the variance of the signal-independent noise component. This thesis employs the Poisson-Gaussian noise model to simulate real world cameras, by combining the effects of shot and read noise.



(a) a=0, b=0 PSNR: 100 SSIM: 1
(b) a=0.4, b=0 PSNR: 10.7 SSIM: $1.8E^{-1}$
(c) a=0, b=0.4 PSNR: 8.21 SSIM: $0.8E^{-1}$
(d) a=0.1, b=0.1 PSNR: 10.7 SSIM: $1.6E^{-1}$
(e) a=0.04, b=0.04 PSNR: 13.6 SSIM: $2.8E^{-1}$
(f) a=0.1, b=0.3 PSNR: 8.53 SSIM: $0.9E^{-1}$

Figure 3.3: Results of adding Poisson-Gaussian noise with different parameters to an image from Set14 [26].

3.1.2. PSNR and SSIM, computing image quality.

Assessing image quality requires considering multiple factors as color, structure, brightness, or contrast. Determining how to evaluate all these aspects is often a subjective and task-specific assignment. Therefore, no universally applicable metrics exists, instead there are numerous metrics, each with its own advantages and drawbacks [10, 16]. Within the domain of image denoising, the two most popular metrics are Peak Signal-to-Noise Ratio (PSNR) [4] and the Structural Similarity Index Method (SSIM) [23]. In this thesis we utilize both metrics to evaluate our experiments' results.

PSNR

PSNR is a metric used to evaluate image quality under several applications [16]. It is easy to compute and is mathematically convenient for optimization. It builds on top of the Mean Squared Error (MSE) calculation, which quantifies the average squared difference between the pixels of an original noise-free image X and of its corresponding denoised estimation $\hat{X}$. MSE is defined for pixels $i = 0, 1, ..., N$ as,

$$\text{MSE} = \frac{1}{N}\sum_{i=0}^{N}(X_i - \hat{X}_i)^2. \tag{3.2}$$

Derived from MSE, PSNR incorporates the maximum pixel value (MAX$_I$) and employs the logarithmic scale:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I{}^2}{\text{MSE}}\right). \tag{3.3}$$

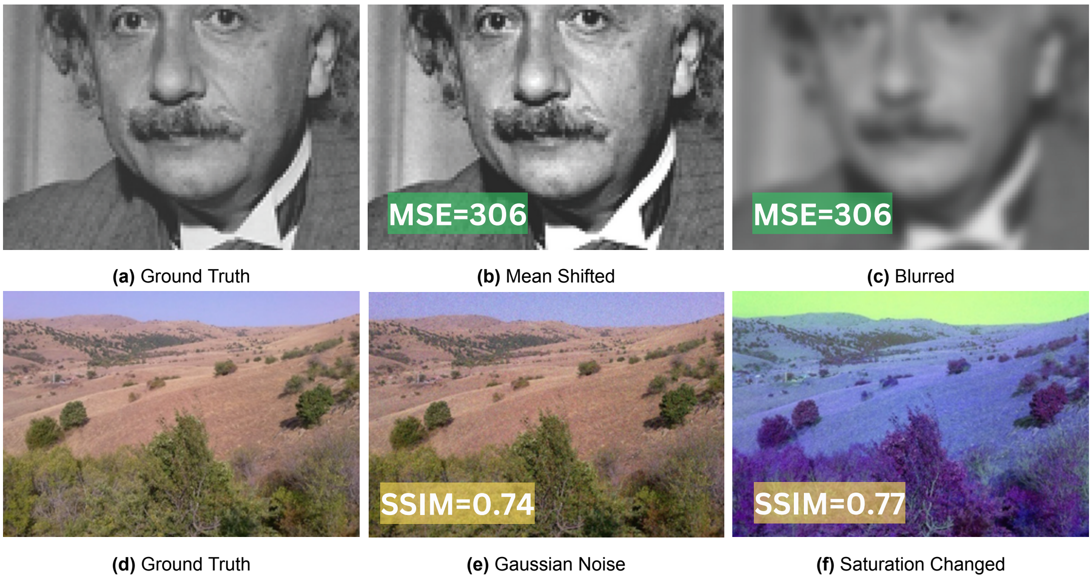
The logarithmic transformation emphasizes even small changes in MSE, making PSNR particularly sensitive to variations in image quality.

SSIM

SSIM is a metric for evaluating the similarity between two images, and is correlated with the human visual perception of image quality [10]. SSIM represents image distortion as a blend of three key elements: loss of correlation, luminance distortion, and contrast distortion. Loss of correlation captures deviations in the relationship between neighboring pixels. Luminance distortion takes into account variations in the brightness or intensity of the pixels. Lastly, contrast distortion addresses discrepancies in the range of pixel intensities.

Selecting an image noise metric.

When it comes to selecting between SSIM and PSNR for evaluating image quality, there exist no strict guidelines, and both metrics present their own challenges (see Figure 3.4). However a dual approach is effective, and leveraging both SSIM and PSNR metrics produces reliable image quality assessments.



(a) Ground Truth **(b)** Mean Shifted **(c)** Blurred

(d) Ground Truth **(e)** Gaussian Noise **(f)** Saturation Changed

Figure 3.4: Challenges with PSNR and SSIM metics. (a-c) PSNR assigns identical scores to distorted images despite obvious quality differences [22]. (d-f) SSIM scores remain similar, even as visual quality varies [14].

3.2. Deep Learning

Within the domain of image denoising and quality enhancement, methods using deep learning approaches have achieved state-of-the-art results [7, 9, 13]. Deep learning uses architectures known as Neural Networks (NNs) to effectively extract important task-specific features from the available data. Furthermore, this thesis investigates transfer learning, a specific technique within the deep learning domain. Here, we will outline the nature of NNs, how they are employed in our thesis, and what transfer learning is.

3.2.1. Neural Networks

Imagine having two data points, x and y, from distinct sets $\boldsymbol{X}$ and $\boldsymbol{Y}$, with unknown underlying relationship $f(\cdot)$,

$$y = f(x), \quad x, y \in \boldsymbol{X}, \boldsymbol{Y}. \tag{3.4}$$

Given many samples of x and y, an NN can approximate $f(\cdot)$.

NNs are intricate functions with numerous parameters. Typically depicted as graphs, nodes represent neurons, mathematical units applying non-linear operations to the inputs, while the edges represent weights, the parameters that scale and alter input data (Figure 3.5).

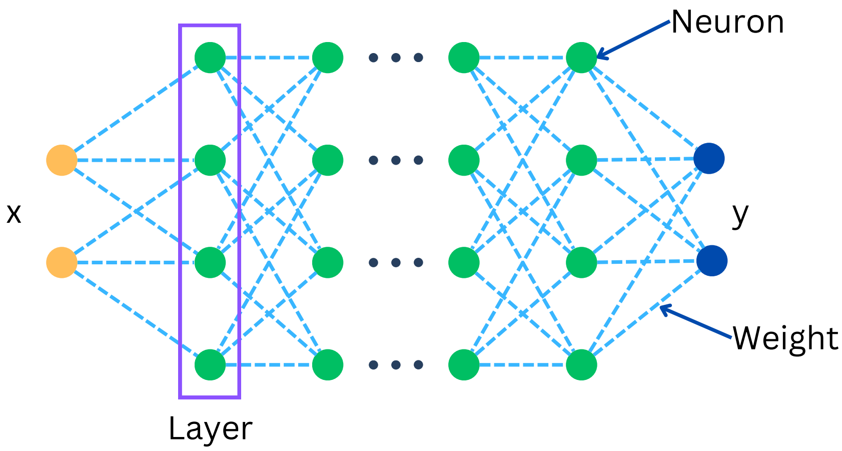


Figure 3.5: Visualization of a Neural Network. The weights adjust the input, while the neurons aggregate inputs and apply non-linear operations. Neurons are commonly organized into layers. The accumulation of layers gives *depth* to the network, inspiring the term ”deep learning.”

In the context of noisy images, consider a clean image X and its noisy variant Y. NNs learn to generate a denoised estimate $\hat{X}$ from the noisy image Y. This learning employs *back-propagation* [20]. In simple terms, back-propagation involves computing the estimation error between $\hat{X}$ and X, and propagating it back across the NN, so that its parameters can be updated to minimize the estimation error.

In essence, deep learning constitutes an optimization challenge: to identify the parameter set θ of an NN $\mathcal{F}$ that minimizes the estimation error. This error is computed with a loss function L, and across a dataset $\mathcal{D}$ of clean-noisy image pairs (X, Y),

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \sum_{(X,Y)\in\mathcal{D}} L(X, \mathcal{F}(Y; \boldsymbol{\theta})). \tag{3.5}$$

3.2.2. U-Net, an encoder-decoder architecture.

The U-Net architecture [19] is a a popular NN design for several imaging tasks. Initially introduced for image segmentation, U-Net architectures are now also extensively employed in image denoising [7, 9, 12, 13, 18]. In this thesis we use this architectures for our experiments. Therefore, it is important to have a general idea of the U-Net structure.

U-Net uses convolutional layers, which are NNs neurons specialized in extracting local image features [17]. The architecture is composed by two main parts: a contracting path, encoder, and an expansive part, decoder (see Figure 3.6).

The encoder captures image features through successive convolutional layers. After each convolutional layer, a pooling operation is applied. Pooling reduce the size of the input by half. This compression creates a densely packed feature representation of the input image.

The decoder converts the abstract features back to the original input image dimensions. Using convolutional layers, the decoder reconstructs the original input by merging features. Instead of pooling, up-sampling operations are applied to expand the input size.

The U-Net's distinctive ”U” shape reflects the symmetry between the encoder and the decoder. This symmetry preserves spatial details while capturing contextual information.

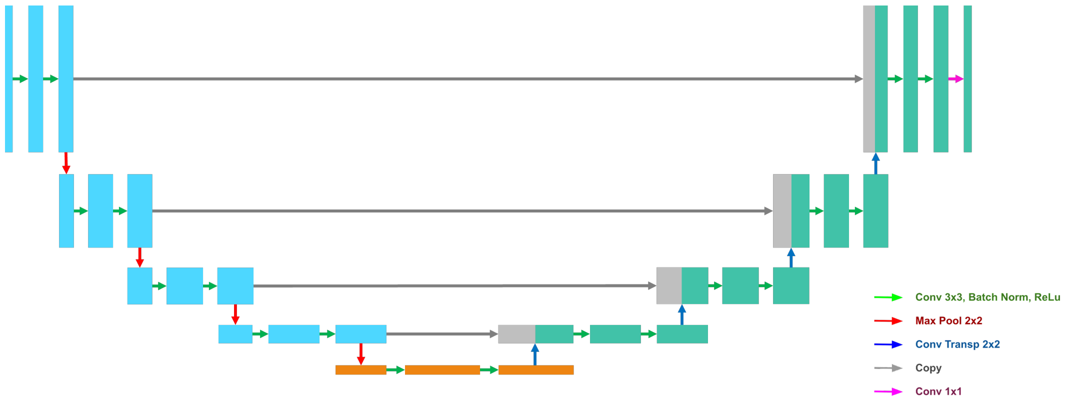


Figure 3.6: U-Net [19] architecture.

3.2.3. Loss functions for denoising.

Several loss functions are used in image denoising. Two common options are the L_2 loss [21] and the L_1 loss [25]. The L_2 loss computes the mean squared difference between the denoised image $\hat{X}$ and the clean image pixels X:

$$L_2 = \frac{1}{N}\sum_{i=0}^{N}(\hat{X}_i - X_i)^2, \tag{3.6}$$

where $i = 0, 1, ..., N$ are the image pixels. Similarly, the L_1 loss measures the mean absolute pixel difference:

$$L_1 = \frac{1}{N}\sum_{i=0}^{N}|\hat{X}_i - X_i|. \tag{3.7}$$

However, both of these approaches have limitations. While L_2 effectively reduces noise, it can overlook blurriness. On the other hand, L_1 is more sensitive to blurriness, but it requires longer reconstruction times. The Charbonnier loss [5] is often employed to address these challenges. This loss behaves like L_1 for large pixel differences and is similar to L_2 for smaller differences. It's formulated using an error term ϵ:

$$L_{Charbonnier} = \sqrt{L_1(\hat{X}, X)^2 + \epsilon^2}. \tag{3.8}$$

In our experiments, we combine the Charbonnier loss with the SSIM loss, where SSIM is computed based on the denoised and ground truth images:

$$L = L_{Charbonnier}(\hat{X}, X) + L_{SSIM}(\hat{X}, X). \tag{3.9}$$

We empirically select this combined approach for our experiments, over using the individual loss functions.

3.2.4. Transfer Learning

Transfer learning is a technique used to train a Neural Network when only a small task-specific dataset is available. First, the network is pre-trained using a large dataset. Then, it is finetuned using the smaller dataset that is specific to the target task (see Figure 3.7).

Consider a real-world analogy. Imagine you want to learn snowboarding, but you have no prior experience with mountain activities. So, you enroll in a snowboarding class. You notice a particular classmate is learning very fast and is already doing the hardest slopes. You ask her how she managed to make this kind of progress with only a couple of lessons. Her response: ”I already knew how to ski really well.” Transfer learning in Neural Networks follows a similar logic. Networks are first trained on a large dataset, learning general patterns. Then, they are finetuned for a specific task, for which we only have a small dataset available. This approach utilizes the knowledge gained from the general task to solve effectively the new task.

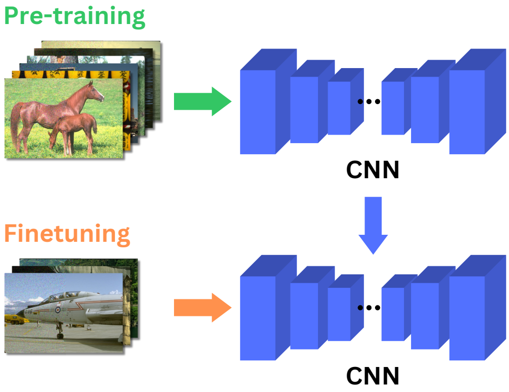


Figure 3.7: Transfer learning scheme. A Convolutional Neural Network (CNN) is first pre-trained on a large dataset with a particular noise distribution. Then it is finetuned with a dataset presenting a different noise distribution, for which only a few samples are available.

Transfer learning is at the forefront of the current NN research [29], since data availability is a recurrent problem in many fields. In this thesis we explore the impact of transfer learning in the image denoising domain. We show the effectiveness of the technique with real-noise image dataset, which are scarce and few. We identify 4 key variables which influence the performance of transfer learning (Figure 3.8), and we investigate them under numerous conditions.

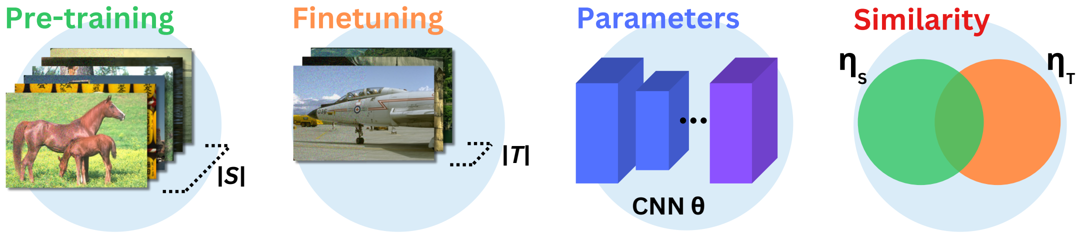


Figure 3.8: Transfer learning variables under investigation in this thesis: (i) pre-training dataset size $|S|$, (ii) finetuning dataset size $|T|$, (iii) network's parameters θ subset which is finetuned, (iv) noise similarity between the pre-training noise distribution η_s and the finetuning noise distribution η_t.

3.3. Datasets

In this section, we present visual examples of the datasets we used throughout the experiments in this thesis.

3.3.1. Toy dataset

The toy dataset is generated combining images from FashoionMnist [24] with a gradient gray background.

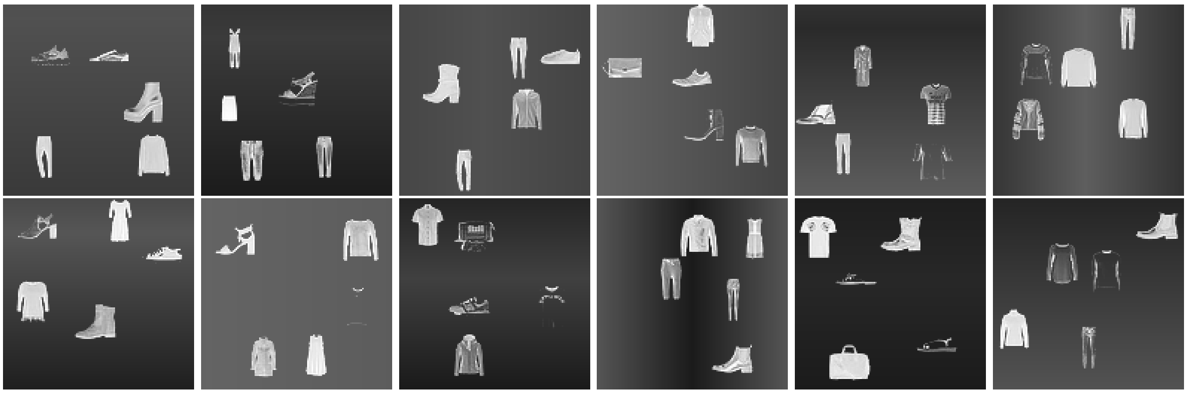


Figure 3.9: Toy dataset

3.3.2. Real image datasets

We use several real image datasets in our experiments. BSD400 [15], BSD68 [15] and Set14 [26] are generic natural-image datasets. Urban100 [11] is a dataset composed by urban scenes images.

Figure 3.10: BSD400

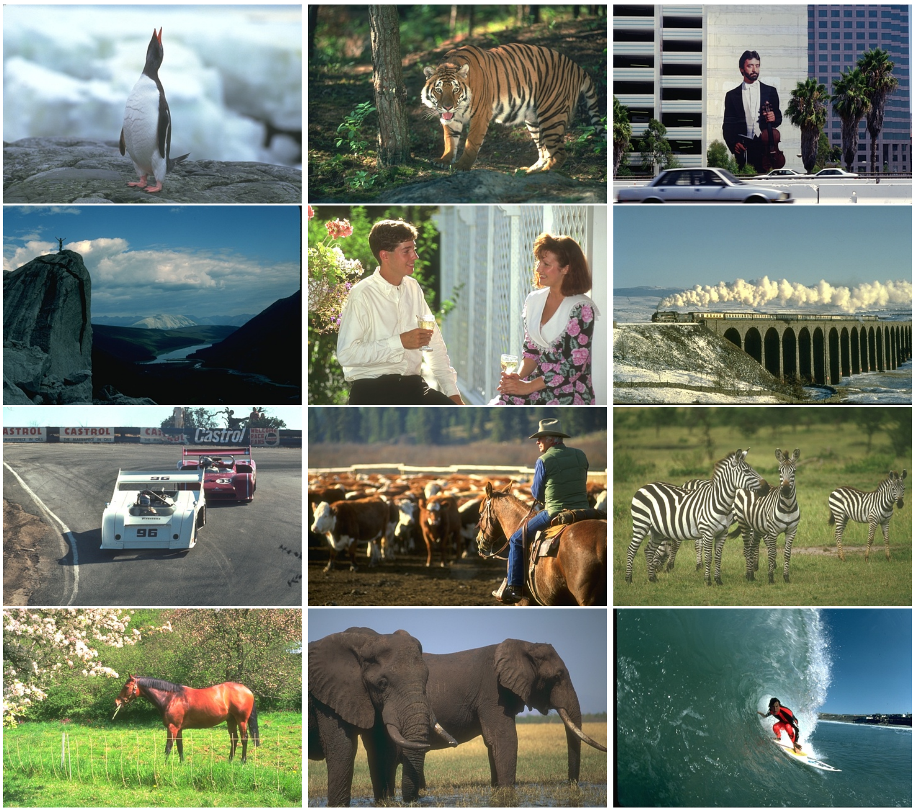


Figure 3.11: BSD68

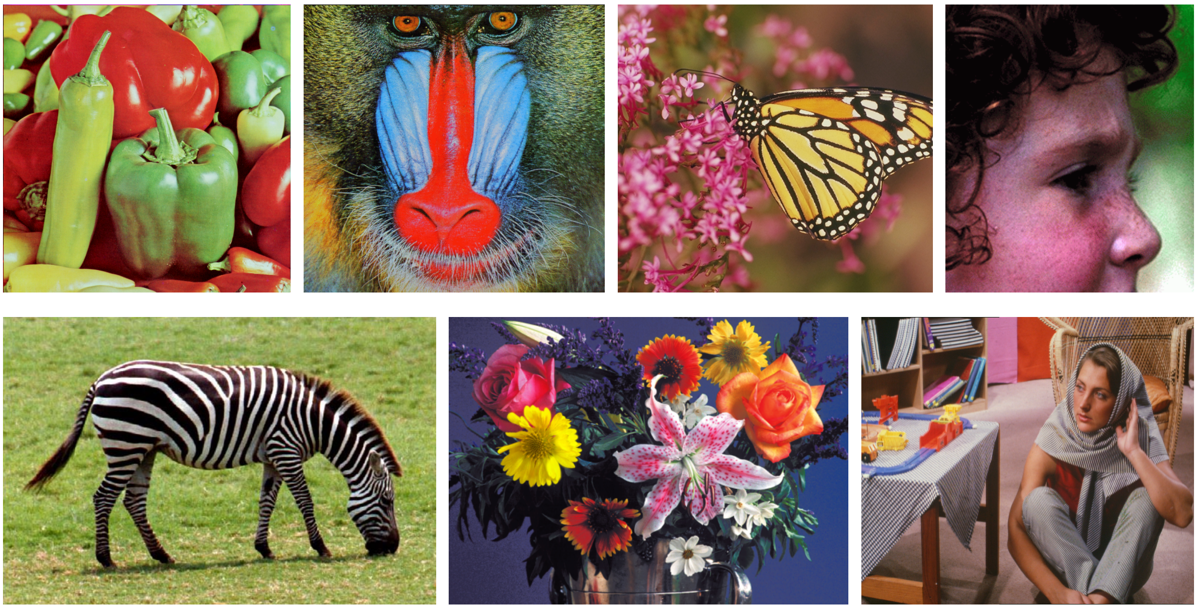
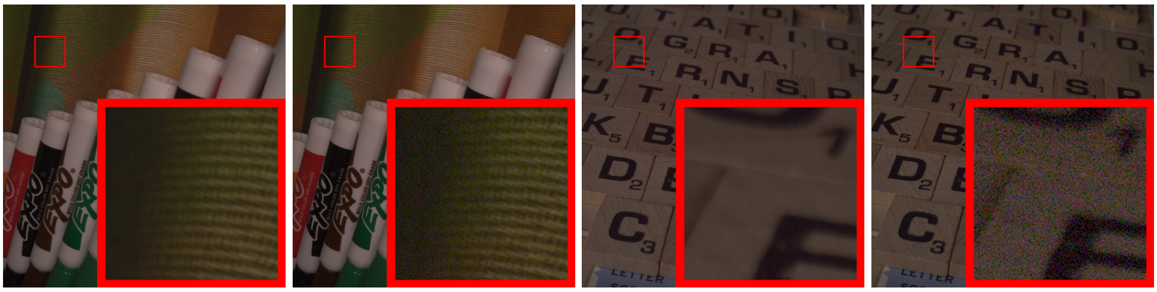


Figure 3.12: Set14

Figure 3.13: Urban100

3.3.3. Real-noise image datasets

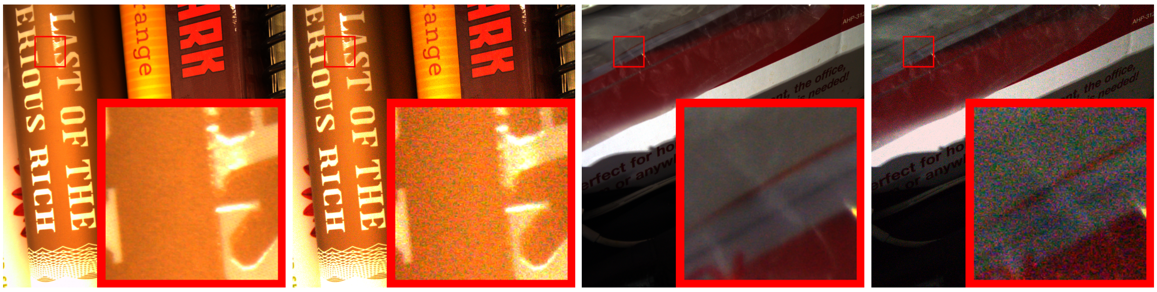
Our real-noise results are validated on subsets of the RENOIR [2] and SSID-Small [1] datasets, selected according to digital camera. Images are here presented as clean-noisy pairs.



(a) ”0045_002”

(b) ”0149_007”

Figure 3.14: LG G4 camera from SSID-Small



(a) ”20160210”

(b) ”20160223”

Figure 3.15: Xiaomi Mi3 camera from RENOIR

Bibliography

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. “A High-Quality Denoising Dataset for Smartphone Cameras”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1692–1700. DOI: `10.1109/CVPR.2018.00182`.

[2] Josue Anaya and Adrian Barbu. “RENOIR – A dataset for real low-light image noise reduction”. In: *Journal of Visual Communication and Image Representation* 51 (2018), pp. 144–154. ISSN: 1047-3203. DOI: `https://doi.org/10.1016/j.jvcir.2018.01.012`. URL: `https://www.sciencedirect.com/science/article/pii/S1047320318300208`.

[3] Nicolas Bahler et al. “PoGaIN: Poisson-Gaussian Image Noise Modeling From Paired Samples”. In: *IEEE Signal Processing Letters* 29 (2022), pp. 2602–2606. DOI: `10.1109/lsp.2022.3227522`. URL: `https://doi.org/10.1109%2Flsp.2022.3227522`.

[4] G. Bjontegaard. “Calculation of average PSNR differences between RD-Curves”. In: *Proceedings of the ITU-T Video Coding Experts Group (VCEG) Thirteenth Meeting* (Jan. 2001).

[5] Pierre Charbonnier et al. “Two deterministic half-quadratic regularization algorithms for computed imaging”. In: *Proceedings of 1st International Conference on Image Processing* 2 (1994), 168–172 vol.2. URL: `https://api.semanticscholar.org/CorpusID:38030033`.

[6] Chen Chen et al. “Learning to See in the Dark”. In: (2018). arXiv: `1805.01934 [cs.CV]`.

[7] Chi-Mao Fan, Tsung-Jung Liu, and Kuan-Hsien Liu. “SUNet: Swin Transformer UNet for Image Denoising”. In: *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2022. DOI: `10.1109/iscas48785.2022.9937486`. URL: `https://doi.org/10.1109%2Fiscas48785.2022.9937486`.

[8] Alessandro Foi et al. “Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data”. In: *IEEE Transactions on Image Processing* 17.10 (2008), pp. 1737–1754. DOI: `10.1109/TIP.2008.2001399`.

[9] Shi Guo et al. “Toward convolutional blind denoising of real photographs”. In: *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[10] Alain Horé and Djemel Ziou. “Image Quality Metrics: PSNR vs. SSIM”. In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 2366–2369. DOI: `10.1109/ICPR.2010.579`.

[11] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. “Single image super-resolution from transformed self-exemplars”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5197–5206. DOI: `10.1109/CVPR.2015.7299156`.

[12] Yifan Jiang et al. *EnlightenGAN: Deep Light Enhancement without Paired Supervision*. 2021. arXiv: `1906.06972 [cs.CV]`.

[13] Yoonsik Kim et al. “Transfer Learning From Synthetic to Real-Noise Denoising With Adaptive Instance Normalization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[14] Zoran Kotevski and Pece Mitrevski. “Experimental Comparison of PSNR and SSIM Metrics for Video Quality Estimation”. In: Jan. 2010, pp. 357–366. ISBN: 978-3-642-10780-1. DOI: `10.1007/978-3-642-10781-8_37`.

[15] D. Martin et al. “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*. Vol. 2. 2001, 416–423 vol.2. DOI: `10.1109/ICCV.2001.937655`.

[16] Yusra Al-Najjar and Der Chen. “Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI”. American English. In: *International Journal of Scientific and Engineering Research* 3.8 (Jan. 2012), pp. 1–5. ISSN: 2229-5518.

[17] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: 1511.08458 [cs.NE].

[18] K. Ram Prabhakar et al. "Few-Shot Domain Adaptation for Low Light RAW Image Enhancement". In: (2023). arXiv: 2303.15528 [cs.CV].

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[20] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[21] Tianyang Wang, Zhengrui Qin, and Michelle Zhu. "An ELU network with total variation for image denoising". In: *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part III 24*. Springer. 2017, pp. 227–237.

[22] Zhou Wang and Alan C. Bovik. "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures". In: *IEEE Signal Processing Magazine* 26.1 (2009), pp. 98–117. DOI: 10.1109/MSP.2008.930649.

[23] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[24] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 1708.07747 [cs.LG].

[25] Xue Yang et al. "Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.2 (2022), pp. 2384–2399.

[26] Roman Zeyde, Michael Elad, and Matan Protter. "On Single Image Scale-Up Using Sparse-Representations". In: *Curves and Surfaces*. Ed. by Jean-Daniel Boissonnat et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 711–730. ISBN: 978-3-642-27413-8.

[27] Kai Zhang et al. "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising". In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155. DOI: 10.1109/TIP.2017.2662206.

[28] Yide Zhang et al. *A Poisson-Gaussian Denoising Dataset with Real Fluorescence Microscopy Images*. 2019. arXiv: 1812.10366 [cs.CV].

[29] Fuzhen Zhuang et al. *A Comprehensive Survey on Transfer Learning*. 2020. arXiv: 1911.02685 [cs.LG].