Assessing Advances in Real Noise Image Denoisers

Clément Bled
Department of Electronic and Electrical Engineering
Trinity College Dublin
Dublin, Ireland
bledc@tcd.ie

François Pitié
Department of Electronic and Electrical Engineering
Trinity College Dublin
Dublin, Ireland
pitief@tcd.ie

ABSTRACT

Recently image denoiser networks have made a number of advances to go beyond additive Gaussian white noise and deal with real noise, such as produced by digital cameras. We note that some of the performance gains reported in the state of the art could potentially be explained by an increase of network sizes. In this paper we propose to revisit some of these advances, including the synthetic noise generator and noise maps proposed in CBDNet, and re-assess them using a simple DnCNN baseline network and thus attempt at measuring how much gains can be attributed to using more modern architectures. In this work, we observe an increase of over +2 dB in denoising performance over our baseline network on the DND real world benchmark. Through this observation, we demonstrate that a smaller networks can offer competitive denoising results when correctly optimised for real world denoising.

CCS CONCEPTS

• **Computing methodologies** → **Image processing**; **Neural networks**.

KEYWORDS

Image denoising, neural networks, real noise

ACM Reference Format:
Clément Bled and François Pitié. 2022. Assessing Advances in Real Noise Image Denoisers. In *European Conference on Visual Media Production (CVMP '22), December 1–2, 2022, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3565516.3565524

1 INTRODUCTION

Image denoising is one of the longest-standing application of image processing and remains, to this date, a key part in many application pipelines, including the in-camera post-processing pipeline, the VFX post-production pipeline, and even the transcoding pipeline of global bandwidth usage.

If early works on Deep Neural Network image denoisers have focused on generic synthetic additive white Gaussian noise (AWGN), it is only since 2019 and the work of Guo et al. [Guo et al. 2019] that the idea of a blind denoising for real images was introduced. Unlike former standards such as Wiener and Wavelet filters, where an estimation of the noise profile needs to be supplied, Guo et al. proposed with CBDNet to train a network for a range of realistic synthetic noise instances and let the network blindly denoise images. Most recent denoising papers have since adopted this approach to achieve near 39-40dB PSNR on popular real-noise online benchmark tables such as DND [Plotz and Roth 2017] and SIDD [Abdelhamed et al. 2018].

Another innovation introduced by CBDNet is the use of a noise map, which provides the network with a tensor of the noise standard deviation at each pixel location. CBDNet trains an ancillary network to predict this noise map tensor. We note that this ancillary network also increases the overall network size, and it becomes then difficult to precisely assess the benefit of this innovation. It may be that the performance increase can be attributed to the networks larger size, rather than the feature being proposed. This is a standard problem with most network innovations and we note that recent networks, such as the transformer-based Restormer network [Zamir et al. 2021a] (26.12 Million parameters) are now over 40 times larger, in terms of number of parameters, than earlier networks, such as DnCNN [Zhang et al. 2017] (0.64 Million parameters). Therefore, in this paper, we propose to carefully investigate the effectiveness of network size and key popular solutions for dealing with real noise, including, the use of noise maps.

For this study, we consider one of the most simple and popular denoising networks, DnCNN [Zhang et al. 2017], as a baseline. This network was originally trained for Gaussian denoising tasks. This will give us an opportunity to observe the effect of combining training data from synthetic, real and Gaussian sources. Our second study analyses the effect of increasing the network size, while maintaining its depth. Following this, we experiment with the effects of including a noise map generation module. Lastly, we compare the DnCNN backbone architecture against more modern solutions, including two novel architectures: resUNext and U2Net.

2 RELATED WORK

2.1 Network Backbones

Neural networks have approached image denoising as an end-to-end problem, directly predicting the denoised image from the input image. The earliest effort using multi-layer perceptron (MLP) [Burger et al. 2012] already came close to surpassing BM3D [Dabov et al. 2007], the reference denoiser that set the benchmark from the late 2000s to early 2010s.

With the advent of larger datasets and more powerful consumer graphic cards, fixed resolution networks such as DnCNN, [Zhang et al. 2017], eventually surpassed BM3D performance in Gaussian denoising tasks. DnCNN also put forward the idea of predicting the residual noise rather than the clean image by adding the input

CVMP '22, December 1–2, 2022, London, United Kingdom

ACM ISBN 978-1-4503-9939-5/22/12.
https://doi.org/10.1145/3565516.3565524

noisy image to the output of the network, which speeds up training and increases denoising performance.

Following this, denoisers have adopted backbone architectures from emerging modern networks such as Resnet (see [Anwar and Barnes 2019; Kim et al. 2019; Maharjan et al. 2019; Ren et al. 2018]), and U-Net style encoder/decoder networks (see [Heinrich et al. 2018; Kim et al. 2020; Zamir et al. 2021b; Zhang et al. 2021]). More recent works have also adopted concepts including attention modules (see RIDNET [Anwar and Barnes 2019; Li et al. 2020]) and subsequently, transformer networks (see U-former [Wang et al. 2022], SWIN [Liang et al. 2021], Restormer [Zamir et al. 2021a]).

The progress made by these networks is notable. On the popular DND benchmark, the performance went from a PSNR of 34.51dB for BM3D, to 38.06dB for RIDNet and up to 39.98dB for U-former.

2.2 Dealing with Real Noise

If it is reasonable in general to assume an additive white Gaussian noise, the precise nature of the noise at hand can be quite different in practice. For instance, camera sensor noise is a mixture of photon noise, dark noise and read noise and is typically approximated as a Poissonian-Gaussian noise, with a signal dependent component (multiplicative) and a signal independent (additive) component [Kokaram et al. 2012; Reibel et al. 2003]. This, combined with the post-processing involved in the in-camera image processing pipeline (ISP) means that the noise is not necessarily additive nor spatially decorrelated.

Thus, at the core of the traditional Wiener and Wavelets denoisers is the concept of *noise profile*. The idea is to measure the expected value of the transform coefficients on patches of real noise, and substract them to real image transform coefficients. For instance, in Wiener, we would measure the noise PSD over a patch of pure noise (eg. over a uniform region of the image).

This noise profile idea turned out not to be directly adapted to the deep learning approach but has still been a source of inspiration in neural methods. For instance, it has been common for AWGN-trained networks to also include the noise standard deviation as an additional channel to the input tensor (see FFDnet [Zhang et al. 2018], DVDNet [Tassano et al. 2019]).

In a traditional machine learning way, it was then shown that the noise profile could be avoided altogether by training a denoising network on enough examples of noise [Guo et al. 2019]. Such a blind denoising approach can in theory predict the noise-free image for any kind of noise degradation.

The key is then to design a noise model that can realistically be used to synthesise any noise type. In CBDNet [Guo et al. 2019], Guo et al. proposed a camera noise generation method, where a clean sRGB image is taken back to a pseudo-raw format by reversing the in-camera functions before adding noise to the image. Signal dependent noise is then added to the pseudo-raw image and the ISP functions are reapplied to obtain the final noisy image. The ISP functions simulated by the synthetic noise generator include Bayer demosaicing, white balancing, colour space conversion and camera response functions. These noisy images are used to train the denoising network (in the case of CBDNet it is a UNet-style network). This approach has been borrowed in many subsequent works [Anwar and Barnes 2019; Liu et al. 2018; Yue et al. 2020].

Another innovation from CBDNet [Guo et al. 2019] was to realise that they could derive from their noise synthesis a per-pixel ground truth value for the noise standard deviation. They refer to this 2D field of the noise intensity as a *noise map* (not to be confused with the noise profile). An ancillary network could then be trained on their synthetic noise to estimate the noise map. Similarly to Multi-Task learning, the idea is that the additional information provided by the ground truth noise statistics could be learned and then exploited by the main denoising network. Noise maps have been used in [Jin et al. 2019; Tian et al. 2020; Yue et al. 2019].

2.3 Remarks

We first note that the blind approach makes a number of interesting assumptions. It presupposes that the training set matches the type of noise for the considered application. There is thus a reliance on the noise generator. We also notice that works tend to simply report the make up of their training set, some of them using the CBDNet noise generator, but none of them discuss or analyse the content of their training set. We found, however, that the CBDNet noise generator suffers from some bias, and that it was possible for denoisers to overfit the CBDNet noise generator. In particular, we observed that the generator tends to suppress the noise for high valued pixels.

Second, noise maps have been adopted in a few papers, however, ablation studies are not presented to clearly demonstrate their benefits in real image denoising. FFDNet reports marginal improvements over DnCNN in Gaussian colour denoising tasks at low to medium levels of noise, with the largest improvements showing for noise of standard deviations of $\sigma = 75$. This level of degradation is much greater than that typically encountered in real photography. CBDNet is assessed only using noise maps, making it difficult to assess whether the networks performance is due to the addition of synthetic noise data in training or the noise prediction sub-network, or both. The ancillary network used to predict the noisemap also bring additional weights to the network which may directly contribute to the denoising process.

3 EVALUATION SETUP

In this paper we propose thus to analyse the impact of the training set used in the literature and to study the effectiveness of training and using a noise map ancillary network as in CDBNet. To make controlled experiments, we will evaluate the methods using a common baseline network. We propose here to work with the DnCNN backbone, as it is a well-known, lightweight network that has shown some success in Gaussian denoising.

3.1 Baseline DnCNN Architecture

We configure the DnCNN network (see Figure 1), to be 20 blocks deep, with each block being made up of a 2D 3×3 convolution, a 2D batch normalisation and a ReLU activation function. The input layer is the only layer to not use a batch normalisation operation, and the output layer does not include an activation function. A skip-connection links the first and final layer of the network via a pixel-wise summation operation. As a result, the network outputs a residual noise image which is added to the original image to produce a clean output.

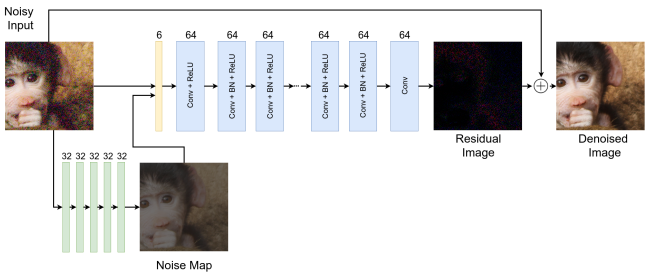

Figure 1: Our DnCNN implementation. The green network is the ancillary noise map prediction network, whose output is concatenated with the input noisy image to form a 6-channels input tensor to the twenty layer DnCNN network (in blue). The output of DnCNN is the residual noise image which, added to the input image, produces the output denoised image.

3.2 Training Regime

Each network is trained for 5000 epochs from sets of real, user generated, and Gaussian datasets. The types of noise used is further explained in Section (4.1). Every epoch, a random 96×96 crop is taken from each image in the training set. An Adam optimiser was used with an initial learning rate of 1×10^{-3} which decayed to 1×10^{-6} using a cosine annealing learning rate function.

3.3 Evaluation Benchmarks

Performances are evaluated using online SIDD and DND benchmarks. This is motivated by the lack of real world-data and the difficulty in comparing real-world denoisers to one another. Unlike Gaussian noise, where a user can generate their own benchmarks with different quantities of noise, the noise here is signal-dependent and cannot be measured using a single value. The use of online benchmarks has become widespread since real-noise denoising has become the primary focus in the literature. This removes the task of creating a sufficiently diverse real noise dataset which can be labour intensive to produce manually. It also allows for using all available real noise datasets for training rather than splitting datasets.

The Smartphone Image Denoising Dataset (SIDD) [Abdelhamed et al. 2018] consists of 1280 noisy image blocks, taken from 40 images, with 32 non-overlapping crops taken from each image. This benchmark is hosted by York University, Ontario Canada.

The Darmstadt Noise Dataset (DND) [Plotz and Roth 2017], consists of 1000 image blocks, taken from 50 images, with 20 non-overlapping crops taken from each image. This benchmark is hosted by the Technical University of Darmstadt, Germany.

4 CHOICE OF TRAINING DATA

Our first set of experiments concerns the choice of training data and the balance between the different noise types. In our experiments, we have three types of noisy data: real noise images, synthetic noise images and Gaussian (AWGN) noise images. We propose to study the effect of training for each of these noise types on the SIDD and DND benchmarks.

Figure 2: Noise samples using CBDNet [Guo et al. 2019] noise generator. Left: original, Right: synthetic camera noise.

Current real world denoisers use a combination of real image datasets in training and synthesised noise. However, little discussion is made on the topic of dataset balance. In [Guo et al. 2019] the authors of CBDNet discuss how training with different types of synthetic noise affects denoising performance. Despite this, there is no consensus on how many real noise images should be included in training sets versus synthesised noisy images. Current networks that use different combinations of real-synthetic training data include GRDN [Kim et al. 2019], RIDNet [Anwar and Barnes 2019], DANet [Yue et al. 2020], AINDNet [Kim et al. 2020] and PT-MWRN [Peng et al. 2020].

4.1 Noise Types

Synthetic Camera Noise. Using images from the DIV2K (800 images) and the MIT-Adobe FiveK [Bychkovsky et al. 2011] (5000 images) datasets, we have generated a training dataset of 3000 images with synthetic noise by adopting the noise generation scheme of CBDNet [Guo et al. 2019]. The images are converted from sRGB to a pseudo-RAW format using randomly selected inverse camera response functions and bayer filters. Signal-dependent noise is applied to the raw images and they are then converted back to sRGB by reversing the initial process. Samples images from this model-based noisy image generation are shown in Figure 2.

Real-Noise. There are only few real-noise datasets available. These are typically obtained by capturing the same scene at different ISO levels, where low ISO levels create the ground truth data and high ISO images create the noisy images. In our experiments we included images from the SIDD dataset [Abdelhamed et al. 2018]. The dataset consists of 320 image pairs taken using smartphones. Ten scenes are recorded under different brightness conditions and illumination temperatures.

Gaussian Noise. Additive white Gaussian noise was also considered so as to assess the possible benefits of including a subset of

Table 1: Impact of training a network on different mixtures of additive white Gaussian noise, synthetic camera noise or real noise. Results are shown for a DnCNN denoiser on the DND Benchmark.

	Gaussian/Synthetic/Real Mix				
	100/0/0	0/0/100	0/100/0	0/91/9	10/83/7
PSNR	34.31	36.97	38.67	39.09	39.02
SSIM	0.8649	0.9248	0.9503	0.9514	0.9519

Gaussian noise to a larger training set and also establish a baseline for a training purely based on Gaussian noise. For this dataset, 1000 unused images from the MITfiveK dataset were used, with Gaussian noise of standard deviation of $\sigma \in [0, 30]$ randomly chosen for each image during the dataloading process.

4.2 Results

We compare the same DnCNN network trained from scratch on the following scenarios: (1) Gaussian noise only, (2) Real noise only, (3) Synthetic noise only, (4) Synthetic and Real noise in a 91/9 split, (5) synthetic, Gaussian and real noise in a 83/10/7 split.

These results for the DND benchmark are shown in Table 1. For each network, an average PSNR and SSIM score is returned. We see that the AWGN trained network (1) performs the worst, with a PSNR of 34.41 dB, while a combination of synthetic and real noise training (4) performs the best in terms of PSNR, at 39.09 dB. Interestingly, the denoiser trained with synthetic noise only (3) significantly outperforms the real-noise trained denoiser, with a difference of 1.7 dB between them. This may be explained by the smaller size of the real noise dataset, as well as the variety of subject matter in the synthetic noise dataset. Another observation we make, is that, although the AWGN dataset contains more than twice the number of images in the real noise dataset, its performance was 2.66 dB worse. This illustrates the ineffectiveness of AWGN training for real photography. Lastly, the incorporation of AWGN into the synthetic-real dataset dropped the final PSNR by 0.07 dB but increased the SSIM result by 0.0005. This suggests that adding Gaussian noise to a dataset may rank the denoisers performance higher in subjective metrics, but note that the margin is too small to definitively make this conclusion.

It is however clear that training on purely synthetic noise yields some overfitting, and that mixing this noise with Gaussian or, better, real noise, improves the overall performance.

5 IMPACT OF THE NETWORK SIZE

As new architectures vary wildly in size and number of parameters, we propose to assess the impact of the network size on the performance of our DnCNN. In the following experiment, we increase the size of our DnCNN denoiser by increasing the number of channels per convolution. Note that doubling the number of channels increases the number of trainable parameters by a factor of four.

Six DnCNN networks are created with 32, 64, 96, 128, 256 and 512 channels per convolution block, with sizes ranging from 0.17 million parameters (DnCNN-32) to 42.51 million (DnCNN-256).

Table 2: Impact of network width on the DnCNN denoising performance on the SIDD Benchmark.

Network Width	PSNR	SSIM	Parameters (Millions)
32 Channels	37.41	0.936	0.17
64 Channels	38.02	0.943	0.67
96 Channels	38.25	0.945	1.53
128 Channels	38.35	**0.946**	2.67
256 Channels	**38.44**	0.945	10.36
512 Channels	38.35	0.945	42.51

Each network is trained on the same mixture of synthetic and real dataset with 91/9 split for 5000 epochs.

5.1 Results

The denoising results for this experiments are presented in the Table 2 on the SIDD benchmark.

Overall, the highest performing network, in terms of PSNR, was our 256 channel network, with a PSNR of 38.44 dB. The 512 channel network, performed worse, despite having over 32 million additional parameters. This is most likely due to overfitting in training. The 512 channel network equalled the result of the 128 channel network at 38.35 dB for the second best performing network. As expected, the 32 channel network, with only 0.17 million parameters gave the lowest PSNR score of 37.41. Despite the smaller size of the network, this score outperforms BM3D, a method once considered to be state-of-the-art.

In terms of SSIM, the 128 channel network scored the highest with a score of 0.946. The 512, 256 and 96 channel networks gave the same, second highest SSIM score of 0.945. The SIDD benchmarking service reports SSIM to three significant figures making it impossible to seperate these scores. Again here, the 32 channel network scores the lowest with a SSIM of 0.936.

From this experiment, we learn that increasing the network size has indeed the effect of increasing the overall performance, with overfitting causing diminishing returns after some point. For comparison, MWCNN [Liu et al. 2018] obtains 39.31dB on SIDD for only 16.14 million parameters, compared to 38.44dB/10.36 millions parameters for DnCNN with 256 channels. Thus, if increasing the network size does indeed have an impact, it is not sufficient to explain the gains obtained with more modern network architectures.

6 EFFECTIVENESS OF A NOISE MAP ESTIMATION ANCILLARY NETWORK

In CBDNet, [Guo et al. 2019] popularised the idea of noise maps, which represent a map of the per-pixel noise standard deviation. Modelling the noise produced by photon sensing as Poissonian and the remaining stationary disturbances as Gaussian, we can indeed model the image noise as a signal dependent process:

$$J = I + \sigma(I)v, \tag{1}$$

where $\sigma(I)$ represents the noise map, I the original clean image and v is Gaussian noise. The noise map $\sigma(I)$ depends on the original

clean signal I as follows:

$$\sigma(I) = \sqrt{\sigma_s I + \sigma_c}, \quad (2)$$

Where σ_s and σ_c represent the standard deviations of the signal dependent and Gaussian independent noise respectively. Unlike typical noise profiles, which use a single value to cover the entire image, this kind of noise map represents the noise intensity per pixel and per channel.

A few works [Guo et al. 2019; Jin et al. 2019; Tian et al. 2020; Yue et al. 2019] have tried to exploit these noise maps by including a small ancillary network that can predict a noise map, which is then fed into the main denoising network, along with the target noisy image. The main idea here is that the additional information provided to the main denoising network will make for a more efficient denoising network. However, no ablation studies have been published to measure the efficacy of using such noise maps, other than showing improvements over existing networks. It may be that the additional parameters provided by the ancillary network assist in denoising directly, instead of providing the main network with information on the noise quality.

To study this, we implement a lightweight ancillary noise estimation network (see Figure 1), composed of ReLU activations and 3×3 convolutions with 32 feature maps in each layer. The noise-estimation network is trained using the noise created when generating our synthetic noisy image dataset, which was saved separately from the final noisy images. As the ancillary network is trained alongside the main denoising network, our custom loss function incorporates four inputs: the ground truth and denoised input image, as well as the ground truth and predicted noise profile. Once the noise map is generated, it is concatenated with the noisy input image to form a 6-channel image-noise map tensor which is subsequently fed into our modified DnCNN network. Other than the larger input, the main model remains the same as our previous DnCNN model.

6.1 Results

In our first experiment, we firstly evaluate whether the use of an ancillary noise map networks yields an added performance boost that is greater than just that of having a bigger network. We train four different networks for this purpose. First we train a 96 channels DnCNN network with noise map ancillary network that is trained on synthetic data to match ground-truth noise map (referred to as 'Trained Noise Map' on Table 3). Secondly, we use the same network but we do not target the ancillary network in training ('Untrained Noise Map'). We achieve this by simply removing the noise map loss function. Thirdly, we train a 97 channel DnCNN network, without an ancillary network ('97 Channel, No Noise Map'). This network corresponds to the same number of parameters as the models with ancillary networks. Lastly, as a final control, we include the 96 channel network without noise maps ('Channel, No Noise Map.).

PSNR and SSIM values for all three scenarios on SIDD benchmark are recorded in Table 3. It is clear from these results that training the network with noise map estimator against ground-truth noise maps provides a significant performance boost in terms of PSNR, with a +0.4dB against similarly sized networks that do not exploit the ground-truth noise maps, i.e. 'Untrained Noise Map' and '97 Channel, No Noise Map'. Note that the increased network size still accounts for a gain of about 0.16dB to 0.17dB. Thus the benefits of using a noise map does not stem from the additional parameters brought by the ancillary network but indeed by exploiting the ground-truth noise-profiles. The differences in terms of SSIM are however not significant.

Table 3: Evaluation of using a noise map for three scenarios on SIDD benchmark. 'Trained Noise Map': the noise map is inferred using an ancillary network and associated to a dedicated noise map loss function as in CDBNet. 'Untrained noise map': the ancillary network is added but without a dedicated loss function. 'No noise map': the size of the original DnCNN is simply increased from 96 channels to 97 channels so as to match the size of the ancillary module.

Network	PSNR	SSIM
96 Channel, Trained Noise Map	**38.67**	**0.949**
96 Channel, Untrained Noise Map	38.26	0.945
97 Channel, No Noise Map	38.25	0.945
96 Channel, No Noise Map.	38.09	0.945

Table 4: Impact of Network Size. PSNR results of our DnCNN trained with different channel widths on the SIDD Benchmark.

	32 channel	64 channel	96 channel	128 channel	256 channel
W/o Noisemap	37.41	38.02	38.25	38.35	38.44
W/ Noisemap	38.09	38.13	38.67	38.38	38.36

We continue the experiment by evaluating noise maps for networks with 32, 64, 128 and 256 channels (see Table 4). It is observed that performance gains are reported over the non-noisemap networks for the 32-channel, 64-channel, 96-channel and 128-channel networks. The most significant performance improvement is noted with the 32-channel network (+0.68dB). This suggest that the noise map can be useful in further improving the performance of smaller denoisers, at low cost. The gains obtained for very large networks are not so clear, with only +0.03 dB at 128 channels and even -0.08dB at 256 channels.

7 CHOICE OF BACKBONE ARCHITECTURE

Following our DnCNN backbone experiments, we also wanted to train larger networks to compare their performance using the same training sets and training regime. The idea here is to explore how much gains can be made by using more modern architectures.

7.1 Proposed New Architectures

We propose two new denoising architectures, based on existing UNet style backbones: ResUnext, which includes a ResNeXt encoder, and U2Net [Qin et al. 2020]. These architectures have shown to be superior to DnCNN in other contexts, and we would like to see how they perform for denoising.

ResUnext. Our proposed ResUnext architecture takes inspiration from existing residual UNet models which incorporate residual connections within CNN blocks, as well as skip-connections between encoder-decoder sections. We have replaced the typical Resnet encoder with a ResNeXt style decoder. ResNeXt [Xie et al. 2017] is a newer implementation of the Resnet architecture which splits convolutional blocks into smaller parallel blocks by equally dividing features between the new blocks. The number of blocks chosen is known as the cardinality of the network. The complexity of the new network is the same as the previous network, but has yielded better performance in ImageNet classification tasks. Our ResNeXt encoder was modelled after a ResNeXt-50 encoder, the smallest model available. Despite this, the size of U-Net based networks are exceedingly large when compared to smaller networks such as DnCNN.

U2Net. Our second proposed network is based on U2Net [Qin et al. 2020], a UNet based network originally designed for image segmentation. In this more complex network, every stage of the encoder and decoder is a small UNet in itself, as shown in Figure 3. The deepest encoder and decoder stages are the exception as the image resolution becomes too small for downsampling. In this case, dilated convolutions are used instead. Unlike typical UNet architectures, each decoder stage produces an output image which is upsampled and subsequently concatenated with every other decoder output. The output images are passed through a final convolution layer to produce the output image. Each upsampled decoder output contributes to the loss function, ensuring that the network is optimised at every level.

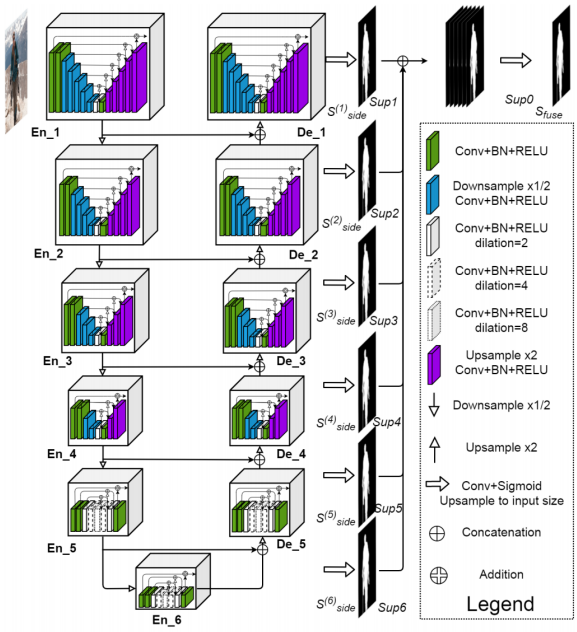

Figure 3: The U2Net Architecture is a UNet styled architecture, where every stage of the encoder and decoder are UNets themselves. The output of each decoder stage is also further reduced to a 3 channel ouput and used in the final image reconstruction, as well as the loss function. [Qin et al. 2020].

7.2 Results

Our results are presented in Table 5 and Table 6 for the DND and SIDD benchmarks respectively. In both cases we include the performance of U2Net and ResUnext in terms of PSNR (dB) and SSIM, as well as some of our previously evaluated DnCNN networks. We also include current state-of-the-art networks, MWCNN and U-former, taken directly from the original author's submissions. Network size is included in terms of millions of parameters. The number of parameters directly affects the RAM used by the network in deployment as well as the model file size.

For the DND benchmark, ResUnext outperforms DnCNN-96 by 0.2 dB/0.0025 SSIM. U2Net outperforms the same network by 0.14 dB/0.0037 SSIM. Although ResUnext (88.6 M parameters) is almost twice as large as U2Net (44 M parameters) in terms of number of parameters, we see only a 0.06 dB difference between them. Furthermore, U2Net outperforms ResUnext in the SSIM benchmark, despite being the smaller network. As all our networks were trained for the same amount of time on the same datasets, this tells us that the size of the network is not always the most pertinent factor in achieving higher performance, although it can help, as shown in our DnCNN size ablation study. The importance of architecture design and optimisation is further underlined by the fact that our best scoring DnCNN network is 59 times smaller than ResUnext and 29 times smaller than U2net, but is only 0.2 dB behind in performance. For both SIDD and DND, our DnCNN implementation outperforms all other submissions with the same backbone (not listed in tables below). In the case of the DND benchmark, our DnCNN-64 network trained on mixed noise using the noise map module outperforms CBDNet, UNet (original) and RIDNet. These network are all significantly larger than ours and were recently considered state of the art in the literature.

U-former's denoising performance (39.98 dB/0.9554 SSIM) shows greater performance than we have been able to achieve thus far. This may be attributed to its transformer backbone or its training material. Like U2-Net and ResUnext, U-former is a large network (50.88 M parameters). This indicates that there are still gains to be made from our larger networks.

Lastly, the network we trained tend to rank lower on the SIDD benchmark. We believe that this may be attributed to the fact that the provided SIDD training set is very similar to the SIDD test set. As such, our training scheme suffers against networks that are purely trained on SIDD data. We feel that this is a problem with the SIDD benchmark and that the DND benchmark is probably fairer in that regard.

8 CONCLUSIONS

In this paper, we have investigated a number of aspects related to targeting real noise in neural networks. First, we have highlighted the importance of training from a combination of real and synthetic noise. Second, we have shown that, a simple DnCNN backbone, with only 0.64 million parameters, when properly trained and optimised, can outperform the denoising performance networks of more complicated recent networks. This includes the original UNet denoiser (30 M parameters) and CBDNet (4.36 M parameters). We

Figure 4: 128x128 crops of denoising results. From Left to Right: Original Noisy, U2Net, ResUnext, DnCNN-96.

Table 5: Popular denoisers evaluated on the DND benchmark. The networks in italic have been trained by us.

Method	PSNR	SSIM	Size (M)
Original Noisy Image	29.84	0.7018	n/a
BM3D	34.51	0.9308	n/a
FFD-NET+	37.61	0.9415	0.85
DNCNN-64 on Gaussian	34.31	0.8649	0.64
DNCNN-64 on mixed	36.97	0.9248	0.64
UNet	38.58	0.9467	30.0
CBDNet	38.06	0.9421	4.36
RIDNET	39.25	0.9528	1.50
DNCNN-64 on mixed + noise map	39.26	0.9530	0.64
DnCNN-96 on mixed + noise map	39.35	0.9530	1.51
ResUnext	39.55	0.9555	88.6
U2NET	39.49	**0.9567**	44.0
MWCNN	39.51	0.9526	16.14
U-former	**39.98**	0.9554	50.88

Table 6: Popular denoisers evaluated on the SIDD benchmark. The networks in italic have been trained by us.

Method	PSNR	SSIM	Size (M)
Noisy Image	23.70	0.480	n/a
BM3D	34.34	0.911	n/a
FFD-NET+	38.27	0.948	0.85
DnCNN-64 on Gaussian	35.32	0.877	0.64
DnCNN-64 on mixed	37.59	0.938	0.64
UNet	38.88	0.952	30.0
CBDNet	34.00	0.868	4.36
RIDNET	38.71	0.952	1.50
DnCNN-64 on mixed + noise map	38.13	0.944	0.64
DnCNN-96 on mixed + noise map	38.67	0.949	1.51
Rexunet	38.06	0.942	88.6
U2NET	38.68	0.951	44.0
MWCNN	39.31	0.956	16.14
U-former	**39.89**	**0.958**	50.88

have also shown that DnCNN-96 is only within 0.63 dB of state of the art network Uformer on the DND benchmark and within 1.2 dB on the SIDD benchmark.

Third, we studied the benefits of estimating noise maps. We found that adding an ancillary noise map estimation network did improve performance (by up to +0.6db). We also established that this was not due to the increased network size. We also observed that the benefits of using noise maps become less noticeable as the network gets larger.

Fourth, we trained two novel larger networks, Resunext and U2Net, of similar size to the highest performing denoising networks available today and found further increases in denoising performance, with U2Net SSIM results being currently within the top 20 DND scores. A comparison with the top-scoring denoisers also shows that there is still a small margin of performance to be gained, which cannot be simply explained by the network size alone. The backbone architecture has therefore still an impact (of about +1 dB).

Last, one issue we face is that the currently available benchmarks are specific to a particular noise profiles. This means that it is quite easy to optimise for a particular benchmark. This is especially true for SIDD as the closeness between its training and testing set probably skews some of the official rankings. We propose that future work should include rethinking the nature of these benchmarks.

ACKNOWLEDGMENTS

This research is supported by Science Foundation Ireland in the ADAPT Centre (Grant 13/RC/2106) (www. adaptcentre.ie) at Trinity College Dublin.

REFERENCES

Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. 2018. A High-Quality Denoising Dataset for Smartphone Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Saeed Anwar and Nick Barnes. 2019. Real image denoising with feature attention. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3155–3164.

H. C. Burger, C. J. Schuler, and S. Harmeling. 2012. Image denoising: Can plain neural networks compete with BM3D?. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Providence, RI, 2392–2399. https://doi.org/10.1109/CVPR.2012.6247952

Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. 2011. Learning Photographic Global Tonal Adjustment with a Database of Input / Output Image Pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing* 16, 8 (2007), 2080–2095.

Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2019. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1712–1722.

Mattias P. Heinrich, Maik Stille, and Thorsten M. Buzug. 2018. Residual U-Net Convolutional Neural Network Architecture for Low-Dose CT Denoising. *Current Directions in Biomedical Engineering* 4, 1 (Sept. 2018), 297–300. https://doi.org/10.1515/cdbme-2018-0072

Yu Jin, Jiayi Zhang, Bo Ai, and Xiaodan Zhang. 2019. Channel estimation for mmWave massive MIMO with convolutional blind denoising network. *IEEE Communications Letters* 24, 1 (2019), 95–98.

Dong-Wook Kim, Jae Ryun Chung, and Seung-Won Jung. 2019. Grdn: Grouped residual dense network for real image denoising and gan-based real-world noise modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. 2020. 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*. 3482–3492.

Anil Kokaram, Damien Kelly, Hugh Denman, and Andrew Crawford. 2012. Measuring noise correlation for improved video denoising. In *2012 19th IEEE International Conference on Image Processing*. IEEE, 1201–1204.

Meng Li, William Hsu, Xiaodong Xie, Jason Cong, and Wen Gao. 2020. SACNN: Self-attention convolutional neural network for low-dose CT denoising with self-supervised perceptual loss network. *IEEE transactions on medical imaging* 39, 7 (2020), 2289–2301.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1833–1844.

Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. 2018. Multi-level Wavelet-CNN for Image Restoration. *arXiv:1805.07071 [cs]* (May 2018). http://arxiv.org/abs/1805.07071 arXiv: 1805.07071 version: 2.

Paras Maharjan, Li Li, Zhu Li, Ning Xu, Chongyang Ma, and Yue Li. 2019. Improving extreme low-light image denoising via residual learning. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 916–921.

Yali Peng, Yue Cao, Shigang Liu, Jian Yang, and Wangmeng Zuo. 2020. Progressive training of multi-level wavelet residual networks for image denoising. *arXiv preprint arXiv:2010.12422* (2020).

Tobias Plotz and Stefan Roth. 2017. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1586–1595.

Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for salient

object detection. *Pattern recognition* 106 (2020), 107404.

Y Reibel, M Jung, M Bouhifd, B Cunin, and C Draman. 2003. CCD or CMOS camera noise characterisation. *The European Physical Journal-Applied Physics* 21, 1 (2003), 75–80.

Haoyu Ren, Mostafa El-Khamy, and Jungwon Lee. 2018. Dn-resnet: Efficient deep residual network for image denoising. In *Asian Conference on Computer Vision*. Springer, 215–230.

Matias Tassano, Julie Delon, and Thomas Veit. 2019. Dvdnet: A fast network for deep video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1805–1809.

Chunwei Tian, Yong Xu, Zuoyong Li, Wangmeng Zuo, Lunke Fei, and Hong Liu. 2020. Attention-guided CNN for image denoising. *Neural Networks* 124 (2020), 117–129. https://doi.org/10.1016/j.neunet.2019.12.024

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. 2022. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17683–17693.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1492–1500.

Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. 2019. Variational denoising network: Toward blind noise modeling and removal. *Advances in neural information processing systems* 32 (2019).

Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. 2020. Dual adversarial network: Toward real-world noise removal and noise generation. In *European Conference on Computer Vision*. Springer, 41–58.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2021a. Restormer: Efficient Transformer for High-Resolution Image Restoration. *arXiv preprint arXiv:2111.09881* (2021).

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. 2021b. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14821–14831.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. 2021. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing* 26, 7 (July 2017), 3142–3155. https://doi.org/10.1109/TIP.2017.2662206 arXiv: 1608.03981 version: 1.

Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual Dense Network for Image Restoration. *arXiv:1812.10477 [cs]* (Dec. 2018). http://arxiv.org/abs/1812.10477 arXiv: 1812.10477.