In this section of the paper, we provide performance on the synthetic datasets and more results on real-image denoising.

1. Identity Module vs. Residual module

Figure 1 shows the difference between our network blocks and resent blocks. It is to be noted here, that resnet employs different block structures in the same network.

As compared to our method, ResNet [5] contains elements such as branched convolutions, strides, batch normalization, flattening, a high number of feature maps, pooling, fully convolutional layer and post-ReLU. Here, it is to be noted that ResNet [5] in its original form is not suitable for image denoising. So, we experimented with a modified version of a ResNet, by setting all the convolution strides to one and removing pooling and fully convolutional layer to make it suitable for denoising purposes. Furthermore, we reduce the depth of the network to 20 layers to make it comparable to our model, DnCNN, and other state-of-the-art networks. The PSNR on BSD68 ($\sigma = 25$) for the modified ResNet network is 16.25 dB without dilation and 16.90 dB with a dilation of three, keeping all the other parameters and training details as ours. This result is very low even compared with decade old methods.

2. inference time and memory cost

The number of parameters in RIDNet is 1.5M (having a model size of 6.0MB) as compared to IERD’s 1M parameter (with 2.7MB model size). Similarly, the inference time taken by our method is 11.42s for the BSD68 dataset, while the time taken by RIDNet is 15.42s. It should be noted here that the time reported also contains loading images, displaying statistics, and saving the results.

3. Datasets

We performed experimental validation on the widely used publicly available three synthetically generated noisy datasets described below.

- Classical images: As a tradition, we first provide a comparison on 12 classical images. The noise of standard deviations (std) of $\sigma_n = 15, 25, 50, 70$ are added to the image.
Figure 2. Denoising quality comparison on a sample image with strong edges and texture, selected from classical image set for noise level $\sigma_n = 50$. The visual quality, i.e., sharpness of the edges on the wings and small textures reproduced by our method is the best among all.

Figure 3. Comparison on a sample image from BSD68 dataset for $\sigma_n = 50$. Our network is able to recover fine textures on the castle.

Figure 4. Denoising performance for state-of-the-art versus the proposed method on sample color images from the dataset in [8], where the noise standard deviation $\sigma_n$ is 50. The image we recover is more natural, contains less contrast artifacts and is closest to the ground-truth.
Table 1. Performance comparison between image denoising algorithms on widely used classical images, in terms of PSNR (in dB). The best results are highlighted with red color while the blue color represents the second best denoising results.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cman</th>
<th>House</th>
<th>Peppers</th>
<th>Starfish</th>
<th>Monar</th>
<th>Airpl</th>
<th>Parrot</th>
<th>Lena</th>
<th>Barbara</th>
<th>Boat</th>
<th>Man</th>
<th>Couple</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM3D</td>
<td>31.91</td>
<td>34.93</td>
<td>32.69</td>
<td>31.14</td>
<td>31.85</td>
<td>31.07</td>
<td>31.37</td>
<td>34.26</td>
<td>33.10</td>
<td>32.13</td>
<td>31.92</td>
<td>32.10</td>
<td>32.37</td>
</tr>
<tr>
<td>WNNM</td>
<td>32.17</td>
<td>35.13</td>
<td>32.99</td>
<td>31.82</td>
<td>32.71</td>
<td>31.39</td>
<td>31.62</td>
<td>34.27</td>
<td>33.60</td>
<td>32.27</td>
<td>32.11</td>
<td>32.17</td>
<td>32.69</td>
</tr>
<tr>
<td>EPLL</td>
<td>31.85</td>
<td>34.17</td>
<td>32.64</td>
<td>31.13</td>
<td>32.10</td>
<td>31.19</td>
<td>31.42</td>
<td>33.92</td>
<td>31.38</td>
<td>31.93</td>
<td>32.00</td>
<td>31.93</td>
<td>32.13</td>
</tr>
<tr>
<td>CSF</td>
<td>31.95</td>
<td>34.39</td>
<td>32.85</td>
<td>31.55</td>
<td>32.33</td>
<td>31.33</td>
<td>31.37</td>
<td>34.06</td>
<td>31.92</td>
<td>32.01</td>
<td>32.08</td>
<td>31.98</td>
<td>32.31</td>
</tr>
<tr>
<td>TNRD</td>
<td>32.19</td>
<td>34.53</td>
<td>33.04</td>
<td>31.75</td>
<td>32.56</td>
<td>31.46</td>
<td>31.63</td>
<td>34.24</td>
<td>32.13</td>
<td>32.14</td>
<td>32.23</td>
<td>32.11</td>
<td>32.50</td>
</tr>
<tr>
<td>DnCNNS</td>
<td>32.61</td>
<td>34.97</td>
<td>33.30</td>
<td>32.20</td>
<td>33.09</td>
<td>31.70</td>
<td>31.83</td>
<td>34.62</td>
<td>32.64</td>
<td>32.42</td>
<td>32.46</td>
<td>32.47</td>
<td>32.85</td>
</tr>
<tr>
<td>DnCNNB</td>
<td>32.10</td>
<td>34.93</td>
<td>33.15</td>
<td>32.02</td>
<td>32.94</td>
<td>31.56</td>
<td>31.63</td>
<td>34.56</td>
<td>32.09</td>
<td>32.35</td>
<td>32.41</td>
<td>32.41</td>
<td>32.68</td>
</tr>
<tr>
<td>IrCNN</td>
<td>32.55</td>
<td>34.89</td>
<td>33.31</td>
<td>32.02</td>
<td>32.82</td>
<td>31.70</td>
<td>31.84</td>
<td>34.53</td>
<td>32.43</td>
<td>32.34</td>
<td>32.40</td>
<td>32.40</td>
<td>32.76</td>
</tr>
<tr>
<td>Our-agnostic</td>
<td>32.11</td>
<td>35.10</td>
<td>33.28</td>
<td>32.31</td>
<td>33.07</td>
<td>31.58</td>
<td>31.80</td>
<td>34.67</td>
<td>32.48</td>
<td>32.42</td>
<td>32.40</td>
<td>32.50</td>
<td>32.82</td>
</tr>
<tr>
<td>Our-specific</td>
<td>32.61</td>
<td>35.21</td>
<td>33.21</td>
<td>32.35</td>
<td>33.33</td>
<td>31.77</td>
<td>32.01</td>
<td>34.69</td>
<td>32.74</td>
<td>32.44</td>
<td>32.50</td>
<td>32.52</td>
<td>32.95</td>
</tr>
</tbody>
</table>

- **BSD68**: Berkely Segmentation Dataset abbreviated as BSD68 ([12]) is composed of 68 images. We provide qualitative and quantitative results for both grayscale images. To generate noisy test images, we corrupt the images by additive white Gaussian noise with standard deviations (std) of $\sigma_n = 15, 25, 50, 70$. 
Table 2. Performance comparison between our method and existing algorithms on the grayscale version of the BSD68 dataset ([8]). The missing denoising results, indicated by “-”, occurs when the method is not trained to deal with the input noisy images.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>31.08</td>
<td>31.32</td>
<td>31.19</td>
<td>31.42</td>
<td>31.74</td>
<td>31.83</td>
<td>31.52</td>
<td>31.86</td>
<td>31.88</td>
<td>31.68</td>
<td>31.81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>28.57</td>
<td>28.83</td>
<td>28.68</td>
<td>28.92</td>
<td>29.04</td>
<td>29.25</td>
<td>29.15</td>
<td>29.05</td>
<td>29.41</td>
<td>29.41</td>
<td>29.30</td>
<td>29.18</td>
<td>29.34</td>
</tr>
<tr>
<td>70</td>
<td>24.44</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
<td>24.43</td>
</tr>
</tbody>
</table>

Table 3. The similarity between the denoised color images and the ground-truth color images of BSD68 dataset for our network and existing algorithms measured by PSNR (in dB) reported for noise levels of $\sigma=15, 25, 50$.

<table>
<thead>
<tr>
<th>Noise Levels</th>
<th>CBM3D</th>
<th>MLP</th>
<th>TNRD</th>
<th>DnCNN</th>
<th>IrCNN</th>
<th>CNLNet</th>
<th>Ours-agnostic</th>
<th>Ours-specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>33.50</td>
<td>-</td>
<td>31.37</td>
<td>33.89</td>
<td>33.86</td>
<td>33.69</td>
<td>33.96</td>
<td>34.12</td>
</tr>
<tr>
<td>25</td>
<td>30.69</td>
<td>28.92</td>
<td>28.88</td>
<td>31.33</td>
<td>31.16</td>
<td>30.96</td>
<td>31.32</td>
<td>31.42</td>
</tr>
<tr>
<td>50</td>
<td>27.37</td>
<td>26.00</td>
<td>25.94</td>
<td>27.97</td>
<td>27.86</td>
<td>27.64</td>
<td>28.05</td>
<td>28.19</td>
</tr>
</tbody>
</table>

Figure 6. A real noisy example from DND dataset ([9]) for comparison of our method against the state-of-the-art algorithms.

Figure 7. A few challenging examples from SSID dataset ([1]). Our method can restore true colors and remove noise.

- CBSD68: Color Berkely Segmentation Dataset abbreviated as CBSD68 ([12]) is composed of 68 images. We provide qualitative and quantitative results for both color images. To generate noisy test images, we corrupt the images by additive white Gaussian noise with standard deviations (std) of $\sigma_n = 15, 25, 50, 70$. 
4. Synthetic Grayscale Image Denoising

In this section, first we demonstrate how our method performs on classical images and then report results on the BSD68 and real datasets.

4.1. Classical Images

For completeness, we compare our algorithm to several state-of-the-art denoising methods using grayscale classical images shown in Figure 2 and reported in Table 1.

In Table 1, we present the average PSNR for the denoised images. Our network is the best performer for almost all classical images except “Barbara”. The reason for this may be the repetitive structures in the mentioned image, which makes it easy for BM3D ([3]) and WNNM ([4]) to find and employ patches with high similarity to the noisy input, hence providing better results.

Subsequently, we depict an example from the classical images. The visual quality of our recovered images, as shown in Figure 2, is better than all others. This also illustrates that our network restores aesthetically pleasing textures. Small and noticeable features restored by our network include the sharpness and the clarity of the subtle textures around the fore and hind wings, mouth, and antennas of the butterfly. Furthermore, a magnified view of the results in Figures 2 for methods such as [3, 15] and [6] show artifacts and failures in the smooth areas.

4.2. BSD68 Dataset

We present the average PSNR scores for the estimated denoised images in Table 2. The IRCNN ([15]) and DnCNN ([14]) network structures are similar, hence produce nearly similar results. On the other hand, our method reconstructs the images accurately, achieving higher PSNR, then completing methods on all four levels of noise. Furthermore, the difference in PSNR between our method and the state-of-the-art techniques at higher noise levels is higher.

For a comprehensive evaluation, we demonstrate the visual results on a selected grayscale image from [12] dataset in Figure 3. In our results; the image details are more similar to the ground-truth details, and our quantitative results are numerically higher than the others. Our method outperforms the second-best method by several orders of magnitude (PSNR is computed in the logarithmic scale). Also, note that the denoising results of other CNN based algorithms are comparable to each other.

4.3. Synthetic Color Image Denoising

For noisy color images, we train our network with the noisy RGB input patches with the corresponding clean ground-truth patches. We only modify the first and last convolution layer of the grayscale network to input and output three channels instead of one channel, keeping all other parameters same as the grayscale network.

We present the quantitative results in Table 3 and qualitative results in Figures 4 and 5 against benchmark methods including the latest CNN based state-of-the-art color image denoising techniques. It can be observed that our algorithm attains an improved average PSNR on all three different noise levels for the color version of the BSD68 dataset ([12]). As shown, our method restores true colors closer to their authentic values, while others fail and induce false colorizations in certain image regions. Furthermore, a close look reveals that our network reproduces the local texture with much fewer artifacts and sufficiently sharp details.

5. Real Image Denoising

We present more example on real image denoising from SSID [1] and DnD [9].

5.1. DnD

Figure 6 shows an image with bricks. The close shows that all other methods have some artifacts while our methods artifacts, which can be seen in the right bottom corner. It can be noted that our PSNR is higher among the competing methods.

5.2. SSID

Our second real image denoising example contains two pictures, as shown in Figure 7. Our IERD and IERD+ have no visual inconsistency and have smoothed the noise while the artifacts are visible for all other methods. The granular structures are present for state-of-the-art methods such as CBDNet and FFDNet, which employ specific techniques to remove the real noise.
6. Summary

In synthetic images case, we have provided ample examples and have shown that our network outperforms classical state-of-the-art denoising algorithms that are intended for use on natural images. Furthermore, we have compared against the current convolutional neural networks, both visually and numerically. Our network gain is about 0.1dB on BSD68 dataset ([8]) and results are visually pleasing.

For the time being, our approach is only applicable to Gaussian noise removal. However, we would like to train our model with different noise types such as Poisson, astronomical etc. and examine its performance on these specific noise types. It should be noted that other state-of-the-art methods, for example, BM3D ([3]), WNNM ([4]) are only applicable to Gaussian noise and may not be readily adapted to handle different noise types.

All CNN approaches performance on images with regular and repeating structures such as “Barbara” is relatively less in terms of PSNR compared to classical denoising methods. This phenomenon is due to the design of traditional denoising methods to exploit the regular and repeating structures. To overcome this issue, either block-matching scheme can be incorporated into our CNN approach or relying on consolidating the outcome of various denoising algorithms with our CNN approach.

References