Image Denoising using Convolutional Neural Networks

Kunhua Lei
Rice University
6100 Main St, Houston, TX 77005
kl61@rice.edu

Iris Wu
Rice University
6100 Main St, Houston, TX 77005
yw110@rice.edu

Danyu Wang
Rice University
6100 Main St, Houston, TX 77005
dw65@rice.edu

Dongwei Li
Rice University
6100 Main St, Houston, TX 77005
dl82@rice.edu

***Abstract*—Image noise can be generated by different types of inevitable factors, like circuitry and sensor of digital camera, or the environment. One of the most challenging parts in processing images is image denoising, and it is essential for applications, for example, improving image qualities and restoration. While there are already lots of proposed solutions to denoising, people are still trying to improve it, especially with high noise level images.**

In this paper, we attempted to build a CNN-based model that perform well in image denoising. This model utilizes convolutional block channel attention and encoder-decoder network with skip connection. We compared our new method with Residual Encoder-Decoder Networks (REDNet) and Pyramid Real Image Denoising Network (PRIDNet) based on Smartphone Image Denoising Dataset (SIDD) training dataset. The new method showed promising improvements in two metrics, Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). In addition, we tested our trained model on single images and Iphone photos to visualize its denoising effect. Github link for the codes is: https://github.com/yw110-1/COMP-576-Final-Project

I. Introduction

Image denoising aims at restoring a clean image from its noisy one, which plays an essential role in low-level visual tasks. In recent years, taking photos with smartphones has become a popular way for people to record their lives. But because of hardware limitations, image denoising still remains one of the most challenging tasks when it comes to processing smartphone images. Nowadays, deep learning techniques have received much attention in the area of smartphone image denoising. There has been considerable research on it, but the denoising model specially designed for mobile phone pictures has not achieved the desired effect. Therefore, in this paper, we attempted to address the denoising task of phone images by using deep learning and convolutional neural networks.

II. Goal

Our goal is to train CNN-based models and adjust network structures on SIDD dataset. We anticipated developing a model that can demonstrate better performances of image denoising by comparing results of different network structures.

III. Data

In order to train our model's image denoising effect for smartphone purposes, we chose the orga-

nized dataset called Smartphone Image Denoising Dataset (SIDD). This dataset consists of 160 image pairs with noisy images and its ground-truth images representing 160 scene instances. For each image, the following is provided: [1]

1) Noisy Raw-RGB image (.MAT). Black Level subtracted, normalized to [0, 1].
2) Ground truth Raw-RGB image (.MAT). Black Level subtracted, normalized to [0, 1].
3) Noisy sRGB image (.PNG). Gamma corrected, without any tone mapping.
4) Ground truth sRGB image (.PNG). Gamma corrected, without any tone mapping.
5) Metadata extracted from the DNG file (.MAT). For example, black and saturation levels, as-shot neutral, noise level function, etc.

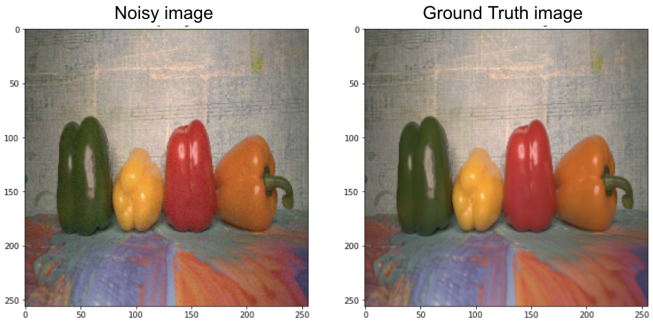

Fig. 1. A pair of sample dataset

We can see from this dataset that it holds a broad representation of different phone images. In addition, the dataset contains high variance environmental factors. Our objective is for our model to learn the generic denoising rules for phone images.

IV. Related Work

By doing research in the image denoising field, we firstly compared two models: REDNet, also known as Residual Encoder-Decoder Architecture, which was published in 2016 by Mao etc; and, PRIDNet, also known as pyramid real image denoising network, was proposed in 2019 by Y. Zhao. The reason we selected these two models is that REDNet is a classic and common baseline model in the image denoising field, and PRIDNet has one of the best performances in denoising.

A. REDNet

REDNet is a classic model for image denoising. It is proposed for indoor RGB-D semantic segmentation and achieved high accuracy.

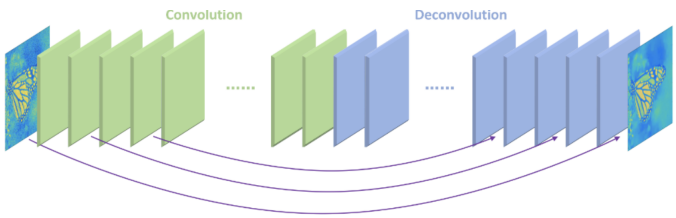

Fig. 2. The overall architecture of REDNet [2]

The proposed framework contains a chain of convolutional layers and symmetric deconvolutional layers, as shown in Figure 2. Skip connections are connected symmetrically from convolutional layers to deconvolutional layers. The method is termed "RED-Net" – very deep Residual Encoder-Decoder Networks. [2]

In order to incorporate the depth information of the scene, there is a fusion structure in REDNet. Firstly, this fusion structure would make inference on both RGB images and depth images. Then, their features would be fused on higher layers by fusion structure. Element-wise summation is performed as the feature fusion method. What's more, in order to optimize the model's parameters more efficiently, a new training scheme which is named 'pyramid supervision' is designed. The pyramid supervision training scheme introduces supervised learning over five different layers, so that it can alleviate the gradient vanishing problem and cope with the problem of gradients vanishing.

B. PRIDNet

In 2019, Yiyun Zhao et al proposed a pyramid real image denoising network (PRIDNet) as shown in Fig. 3. The method includes three modules: [3]

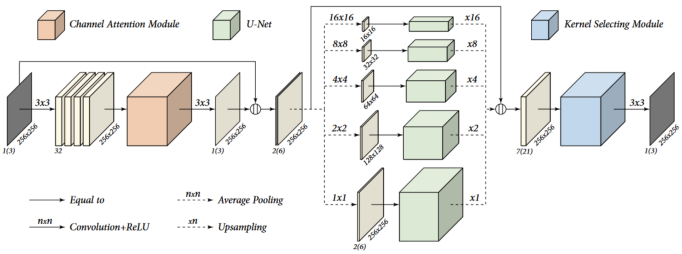

Fig. 3. The structure of PRIDNet [3]

- *Channel attention:* Channel attention mechanism is utilized to adjust the channel importance of input image data.
- *Multi-scale feature extraction:* This part includes an average pooing with given kernel size and a U-Net architecture.
- *Kernel Selecting Module:* In order to choose size-different kernel for each channel within concatenated multi-scale results, a kernel selecting module is introduced.

V. Our Model

In order to find a new solution, we developed a novel method by taking advantage of previous two models. Figure 4 shows the overall structure of our model. It can be divided into two main parts. The first part is the channel attention module. The input layer is connected to a convolutional block, which contains 4 convolutional layers and uses Relu as activation function. Then we used an attention mechanism to put weights on each channel.

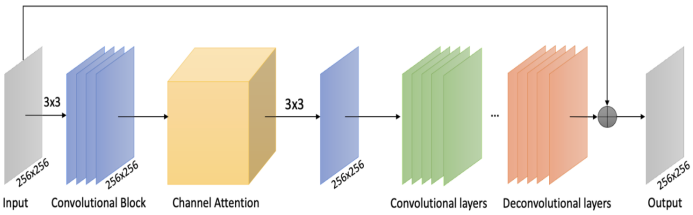

Fig. 4. Architecture of Our Model

As Figure 5 shows, in the channel attention block, we implemented a global average pooling, and passed it to 2 fully connected layers. We get a trainable vector with the number of channels, which is the attention we will put on each channel. And this weight vector will be multiplied with the input to obtain an output.

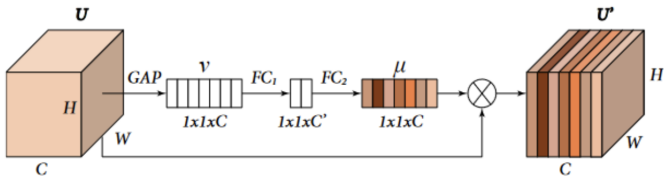

Fig. 5. Channel attention module. The GAP denotes the global average pooling operation. FC_1 has a ReLu activation after it, and FC_2 has a sigmoid activation after it. [3]

The second part is the encoder and decoder module. It contains 5 convolutional layers for the encoder and 5 deconvolutional layers for the decoder (we also increase the number of layers in our experiments section). The convolutional layers extract the feature, which is the primary components of objects in the image. After forwarding through the convolutional layers, the noisy input image is converted into a "clean" one. The role of deconvoutional layers is to recover the details of the image, which output a cleaned version of the images. As the convolutional network goes deeper, however, too much detailed information may be lost, making deconvolution weaker in recovering them. Skip connection is added to pass the feature map of convolutional layer to its corresponding deconvolutional layer.

- **Attention mechanism**: It is well known that attention plays a significant role in human perception. Human's eyes will focus and emphasize the most important part when they capture a view of scene. In artificial neural networks, attention mechanism is a technique that mimics human's cognitive attention. This mechanism will add more weight on some parts of the input data while diminishing other parts. [6] In our image-denoising model, we adopt a channel attention module. The motivation is to devote more training focus to the important parts of an image.

- **Residual Encoder-Decoder Network**: The RED-Net module is the main part of our model. Instead of all fully convolutional networks, we used a symmetrically equal-number of convolutional and deconvolutional networks. Moreover, we also add skip connections from a convolutional layer to its corresponding mirrored deconvolutional layer.

VI. Experiments

A. Training

We trained three models using Google Colab GPU. We used the function 'train_test_split()' in scikit-learn to randomly select 80% of the SIDD as the training data and 20% as the testing data. The models were trained by Adam Optimizer with 0.0001 as the learning rate. The loss function was calculated using the root mean squared error between the ground truth image and predicted image.

When training the REDNet, we tried changing the number of convolutional layers and deconvolutional layers, to figure out whether more layers could help to improve the denoising effect. We set the number as 5, 9, and 15. Also, we have compared our results with original REDNet and PRIDNet performances.

B. Testing

Testing Single Image Prediction To visualize the denoising performance, we used several noisy images as inputs, and generated predicted images using our model. As Figure 6 is shown, we can clearly see that the predicted images of the eye-closeup and rail train images have much less noise than original images. It intuitively proved that our model has a good denoising performance.

Definition of Metrics To quantitatively evaluate the performance of our result, we use the value of two metrics, Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). PSNR quantifies reconstruction quality for images. The higher the PSNR, the better the quality of our predicted image. It is defined via the mean squared error (MSE). Given a noise-free mxn monochrome image I and its noisy approximation K, MSE is defined as [5]

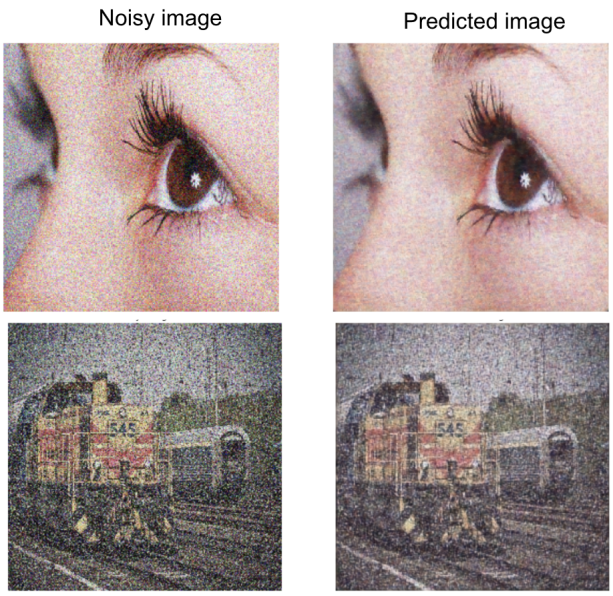

Fig. 6. Test Images Examples

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

The PSNR (in dB) is defined as

$$\begin{aligned}\text{PSNR} &= 10 \cdot \frac{\text{MAX}_I^2}{\text{MSE}} \\ &= 20 \cdot \frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \\ &= 20 \cdot \text{MAX}_I - 10 \cdot \text{MSE}\end{aligned} \quad (2)$$

where MAX_I is the maximum possivle pixel value of the image.

SSIM measures the structural similarity between the predicted images and ground truth images. It is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$ is: [4]

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

where:

μ_x is the pixel sample mean of x;
μ_y is the pixel sample mean of y;
σ_x^2 is the variance of x;
σ_y^2 is the variance of y;
σ_{xy}^2 is the covariance of x and y;
$c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ are two variables to stabilize the division with weak denominator;
L is the dynamic range of the pixel-values;
$k_1 = 0.01$ and $k_2 = 0.03$ by default.

As defined above, our experiment obtained the value between ground-truth image and predicted image. We used 'peak_signal_noise_ratio' from 'skimage.metrics' in our code. The data is shown in Table I and Table II.

To further observe the effect of our trained model, we tested it on some noisy photos taken by iPhone XS. The parameter of the iPhone camera was set to a Wide Camera with 26 mm $f1.8$ and a focus length of 239 mm to make some noise in purpose. We can only see slight changes by comparing the noisy images and predicted images as shown in Figure 7. It might be because the noisy images had been processed by iPhone when they were taken.

VII. Results

We provided the results of our experiments based on single image prediction. Table I showed PSNR and SSIM in different trainings, original REDNet with various layers, original PRIDNET, and our method, which is termed 'REDNet_attention', with 5 layers (red color). Although we had increased the number of encoder-decoder layers for our method, which is meant to deepen the convolution network, we found that it could only slightly improve the testing scores (PSNR and SSIM). The data is not showed in our report. However, on the other hand, the time for training increased a lot.

According to Table I, our model has about 7.2 increase in PSNR and 0.18 increase in SSIM from original pairs, which means the predicted image using our model can have a better quality and structural similarity to the real image. Also, adding layers can increase the quality of image by denoising as well.

Figure 8 shows the relationship between epoch and PSNR, SSIM. We can see that as epoch in-

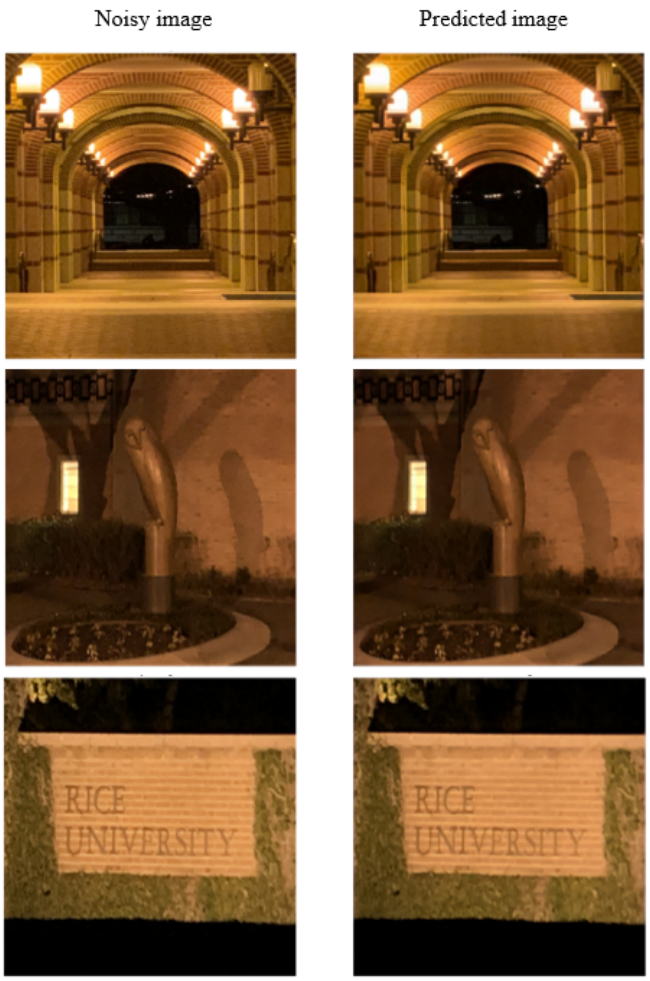

Fig. 7. Test trained model on iPhone photos. The model was trained with 30 epochs

TABLE I
Evaluation of Performances on Models

Model	PSNR	SSIM
Original X-y pairs (no model)	26.3779	0.6000
REDNet_5layers	32.1988	0.7725
REDNet_9layers	32.7721	0.7728
REDNet_15layers	33.1718	0.7738
REDNet_attention_5layers	33.5744	0.7785
PRIDNet	33.3105	0.8049

TABLE II
PERFORMANCES WITH DIFFERENT EPOCHS

REDNet_attention_5layers	PSNR	SSIM
5 epochs	32.0971	0.7662
10 epochs	32.9155	0.7815
30 epochs	34.4959	0.8098
60 epochs	33.9822	0.8389

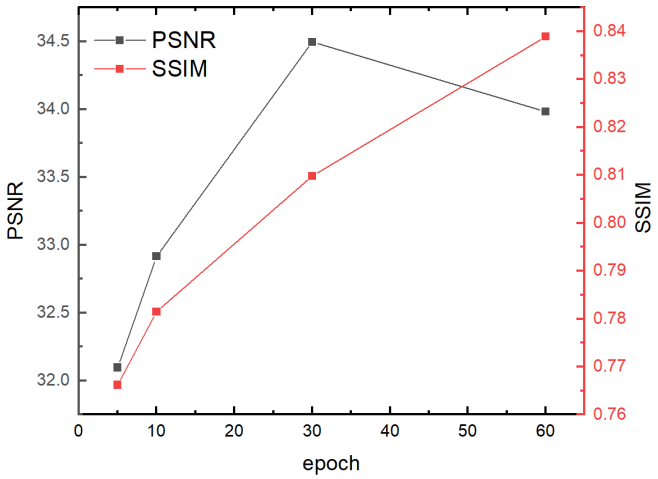

Fig. 8. PSNR and SSIM versus epoch

creases, PSNR and SSIM also increase. It seems that the two metrics tend to stabilize when training epoch is above around 50. In our experiments, we tried 80 epochs with Google GPU. But the whole training took too long to finish (due to Google GPU usage limit). Figure 9 visualized the difference between model trained with 30 epochs and 60 epochs. The output image in right below is slightly clearer than the one in right above.

VIII. CONCLUSION

In conclusion, we built a new CNN-based model for image denoising. The model contains a channel attention mechanism and an encoder-decoder architecture. Through preliminary training and testing, the model showed a much better performance than REDNet. PSNR and SSIM both increased compared to previous reported methods (REDNet and PRIDNet). Our method is lighter-weight compared to PRIDNet but obtained a better result in PSNR.

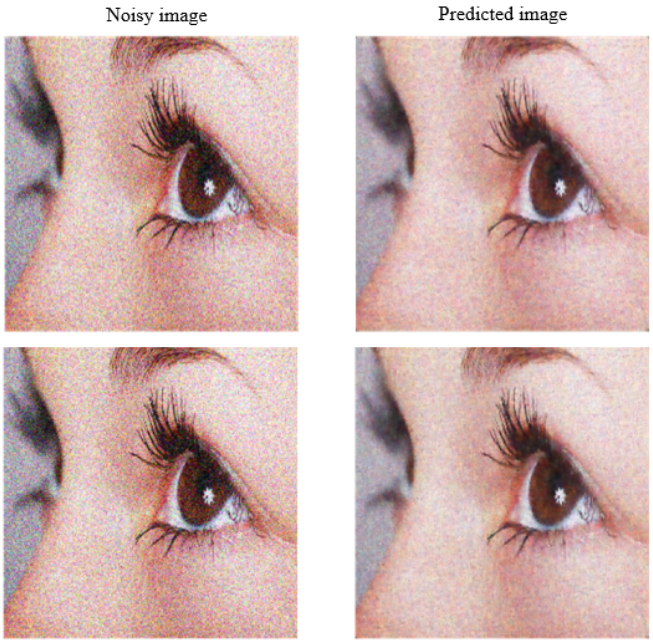

Fig. 9. Comparison of 30 training epochs (above) and 60 training epochs (below)

However, we found some limitations to our model and training process. Firstly, our model shows a higher PSNR, but the SSIM value is not so higher than PRIDNet, which means although our model can reconstruct images with higher quality, it cannot perform a good similarity performance with the ground truth image. Secondly, since we only used basic GPU from Google Colab, our testing results might be undervalued.

Therefore, in the future, we think there are some potential improvements or considerations on training and testing our model. Training with larger datasets can certainly improve the training and testing. For example, SIDD full dataset includes roughly 30,000 images, which is a lot more than the small dataset that we used. Also, from Table II, we see that adding epochs is able to improve both PSNR and SSIM values. Thus, if we have a more powerful GPU, we can train with large numbers of epochs (e.g. 100 epochs, 1000 epochs,

. . .). In addition, k-fold cross validation is another useful technique to prevent overfitting and estimate a model.

REFERENCES

[1] York University, Smartphone Image Denoising Dataset, https://www.eecs.yorku.ca/ kamel/sidd/.

[2] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang. "Image Restoration Using Convolutional Auto-Encoders with Symmetric Skip Connections". 30 Aug. 2016, doi:10.48550/arXiv.1606.08921.

[3] Yiyun Zhao, Zhuqing Jiang, Aidong Men, Guodong Ju. "Pyramid Real Image Denoising Network". 2019 IEEE Visual Communications and Image Processing (VCIP), 2019, pp. 1-4, doi: 10.1109/VCIP47243.2019.8965754.

[4] "Structural Similarity" Wikipedia, Wikimedia Foundation, 6 Nov. 2022, en.wikipedia.org/wiki/Structural_similarity.

[5] "Peak signal-to-noise ratio" Wikipedia, Wikimedia Foundation, 22 Aug. 2022, en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

[6] "Attention (machine learning)" Wikipedia, Wikimedia Foundation, 4 Dec. 2022, en.wikipedia.org/wiki/Attention_(machine-_learning).

[7] Abdelrahman Abdelhamed, Lin S., Brown M. S. "A High-Quality Denoising Dataset for Smartphone Cameras", IEEE Computer Vision and Pattern Recognition (CVPR), June 2018.

[8] Abdelrahman Abdelhamed, Timofte R., Brown M. S., et al. "NTIRE 2019 Challenge on Real Image Denoising: Methods and Results", IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), June 2019.