Unsupervised Training of Denoising Networks

Alexey Kovalenko and Yana Demyanenko
Southern Federal University, Rostov-on-Don, Russia
alexey.s.russ@mail.ru, demyanam@gmail.com
https://www.sfedu.ru/

Abstract. This work explore approach for image denoising of received by CMOS sensor. Proposed pipeline solves the problem of unsupervised training neural network architectures for image denoising which uses datasets without clean data. This approach bases on theoretical background about image restoration proposed by Nvidia researchers. We implemented custom denoising neural network architectures using specifics of noise distribution. Networks are trained on custom images collection.

Keywords: Image denoising · Unsupervised learning · Neural networks · Learning image denoising

1 Introduction

Noise reduction is common problem in computer vision. Any image captured by CMOS sensor contains noise. This noise appears in useful signal due to errors in reception of optical radiation by sensor. Clear image signal will be denoted by $I$, and noise component denoted by $\alpha$. Assuming that the process of noise occurrence is an absolutely random process from the distribution of $P$. Then the matrix of the final image can be denoted by formula (1).

$$\tilde{I} = I + \alpha, \alpha \sim P$$ (1)

Since the error in receiving optical signal depends on physical device of the CMOS sensor, that for each model of the sensor there will be a unique distribution of $P$, which generates the noise component of the signal.

The goal of this work is approximation of the function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by neural network, which has the following property:

$$\forall \tilde{I} \Rightarrow \phi(\tilde{I}) = I$$ (2)

For build of approximation of the mapping $\phi$, the neural network $f$ will be trained to solve the following optimization problem:

$$\min_w \|f(\tilde{I}, w) - I\|_{L_2},$$ (3)

where $w$ are the neural network weights $f$, $L_2$ is Euclidean norm.
2 Related Works

Significant contribution to the denoising problem by neural networks is made by the work of Nvidia researchers. This work is titled Noise2Noise: Learning Image Restoration without Clean Data [1]. The main idea of this work is using representation of noise as a composition of a clear signal and unique noise received at different points in time to train a neural network to restore a clean image signal. A noisy image with the noise component $\alpha_1$ is forward to neural network and the network is required to predict the same image, but with the noise component $\alpha_2$. Assuming that the noise was obtained randomly, the neural network is not able to predict it, and therefore, during training, the neural network seeks to reconstruct the image with some losses. The disadvantage of this approach is the use of only a pair of images of one scene during training, which can result in large losses of the useful signal during network operation.

There is also a lot of researches touch upon problem of training noise reduction networks on datasets containing images without a noise component. An example of this approach is the work of engineers from Google which titled Unprocessing Images for Learned Raw Denoising [2]. In this work, the authors will train the network using the standard error function $L_1$. And tested this network on a Darmstadt Noise Dataset [3].

Darmstadt Noise Dataset datasets contains pairs of images. Each pair consists of an image taken with correctly selected camera parameters for shooting and an image having noises arising from incorrect parameters. Networks trained on such data sets do not solve the problem of suppressing noise arising from the CMOS sensor of a camera even in the most correctly selected shooting parameters.

Thus, for training a network focused on denoising from a specific CMOS sensor, it becomes necessary to collect data and develop a method for training a network on them.

Together with Noise2Noise work [1], there are state-of-the-art approaches such as Noise2Void [4], Noise2Self [5] and Deep Image Prior [6], contributing to the field of unsupervised learning of denoising neural networks.

3 Dataset

3.1 Collecting Dataset

Images for training dataset were obtained using a certain device, Apple iPhone X, which has a camera consisting of two sensors, with the characteristics given in the following source [7].

From this device a series of RAW images of seven scenes with different lighting and color component were obtained. A scene made by shooting an static picture of the real world, getting the matrix (1). To do this, the device was fixed on a tripod in a stationary state and the shooting process was started from a wireless device, thereby obtaining a set of images $\{I^q_k\}_{k=1}^N$, where $q$ is the sequence number
of the scene being shot. For each series ISO, focus and color temperature values were fixed when shooting.

Each scene contains about 14 photos. Total number of frames from all sets is 95 images.

The maximum number of images in one series is limited to 20 frames because CMOS sensor heats with long time the shooting process continues and additional signal distortions appear due to thermal effects. Due to this effect of which the pixel-by-pixel correspondence of frames in the series to each other is lost.

The result is a dataset consisting of 125 images taken on one sensor.

Also, to compare the results, 6 series were made, shot by next web camera in the resolution of 1920 × 1080. Short videos were shot and divided into frames. Total number of frames in these series was 811.

### 3.2 Analysing Dataset

During the shooting process, slight distortion of the frame may occur when the photosensor is heated. Or imperceptible displacements of the device during shooting, as well as a change in external conditions, such as lighting or movement of objects in the frame, are also possible. For an collected set of images, a shift value of more than one pixel between frames of the same series is already critical.

To analyze the quality of the obtained series of images, the distributions of $e^q$ (5) deviations of each image from the average over all images from the series $\hat{I}^q$ by the Euclidean metric.

$$\hat{I}^q_{i,j} = \frac{\sum_k \tilde{I}^q_{k,i,j}}{N}, \text{ where } N \text{ is the number of frames in series } q \quad (4)$$

$$e^q_k = \|\hat{I}^q - \tilde{I}^q_k\|_{L_2} \quad (5)$$

Additionally, the series is normalized by metric $L_1$:

$$e^q_k = \frac{e^q_k}{\sum_i (e^q_i)}$$

Then, using the values of the sample $e^q$ (5), we plot the density curves of the normal distribution $N(\mu, \sigma)$ with the parameters:

$$\mu^q = \mu(e^q), \quad \sigma^q = \sigma(e^q) \quad (6)$$

Thus, the mathematical expectation of $\mu^q$ (6) shows how near the average image of $\hat{I}^q$ (4) to the image theoretically obtained from a pure signal $I^q$:

$$\hat{I}^q \xrightarrow{\mu^q \to 0} I^q \quad (7)$$

And also, the near to 0 the value of the standard deviation of the sample $\sigma^q$ (6), the more images from the series there is a signal of static scene, fixed at time $t = t_0$, that is, how much the scene is unchanged between frames.
Fig. 1. Series of images with color histograms, the parameter $\sigma$ of this series is 0.00608

For example, the deviation parameter $\sigma$ (6) may be quite large with different illumination of the same scene between frames (see Fig. 1).

For demonstrate the various values of the deviation parameter, a graph is constructed (see Fig. 2).

Fig. 2. Normal distribution density plots for series shot under natural light conditions

When choosing data for training neural networks, the condition is set that the parameter $\sigma$ (6) should not exceed the value of 0.007.

4 Method

In this work we produce a novel method for training denoising neural network architectures. This approach combine traditional supervised method for training
autoencoder neural network architectures and approach produced by authors of Noise2Noise work [1].

When we have image series of one scene obtained in different time (see Sect. 3), we can average this image series and made approximation of expected clear image. Obtained mean image can be used for train the denoising neural network as the expected output signal $y'$ in solving of loss $E$ minimization problem (9). An input signal $x$ can be taken random image from series under consideration. But in this case value $y'$ consists noise component from input signal $x$. If input image has been excluded from series before averaging step, then we made approximation of clear image without noise component of input signal and this setting is consistent with theory of Noise2Noise work [1]. This ground truth signal $y'$ are constructed using the formula (8). Thereby, when we train denoising network to predict signal $y'$, it train to extract additional features from input signal by Noise2Noise approach logic for achieving more denoising quality. Scheme of this approach also is shown (see Fig. 3).

\[
X_i = \bar{I}_i, \quad y = \frac{\sum_{k=1, k \neq i}^{N} \bar{I}_k}{N-1} \tag{8}
\]

\[
\min_w E(f(x), y') \tag{9}
\]

![Fig. 3. Scheme of training pipeline](image)
5 Architecture

5.1 Base Architecture Block

To build architectures in this work for test proposed approach, the blocks from the ResNet [8] architecture were taken and modified. From ResNet blocks batch normalization layers has been excluded. The scheme of these blocks is shown (see Fig. 4).

![Fig. 4. Detailed scheme of modified ResNet block](image)

5.2 Architecture with Frequency Components Splitting

The noise in question (1) belongs to the high frequency component of the image signal. An example of decomposition of an image signal into high and low frequencies using the Butterworth filter is shown (see Fig. 5).

Taking into account such a distribution of the noise component, an architecture was developed with the separation of the signal into high-frequency and low-frequency signals and their further independent processing by the convolutional layers of the network. This scheme is shown (see Fig. 6).

Implemented architecture (see Fig. 6) consists of a layer of frequency signal decomposition and sequential processing of parts of the input signal using convolutional blocks. Then the signals from blocks with are concatenated by channels and processed by a 3D convolutional layer to extract high-level features with interchannel dependence.

This architecture has 511390 of trainable parameters and the weight of this model is 2 megabytes when using single precision FP32 numbers to store the parameters.
Fig. 5. From left to right: original part of the image, 2D Fourier spectrum for the grayscale image, rendering of the Butterworth filter, low-frequency component of the image, high-frequency component of the image.

6 Training

In the experiment we train architecture with frequency components splitting (see Sect. 5.2) by proposal approach (see Sect. 4) on our dataset (see Sect. 3).

To use all frames from a series of images, one frame from a series was use to the network input, and a series averaging without an input frame was used as the expected output signal (see Sect. 4). Based on the sample \( \{ \tilde{I}_k \}_{k=1}^N \), the input \( X \) and the expected at the output of the network \( y \) are constructed using the formula (8).

Denoising model was trained on random cropped images and using the following augmentation: rotations by 90, 180, 270° and flips.

For denoising models training was implemented pipeline based on PyTorch framework. As optimizer for was models training used radaM optimizer [9].

7 Inference

On inference input image unfolds on grid of crop. Network forwards on each crop with test-time data augmentation with transforms from training [10]. Result image computed by formula (10), where \( T_i \) - augmentation transform, \( T_i^{-1} \) - inverse transform for correspondent transform \( T_i \).

\[
I_{result} = \frac{\sum_{n=1}^{8} T_i^{-1}(f(T_i(x)))}{8} \tag{10}
\]
Fig. 6. Detailed architecture diagram of denoising network with frequency decomposition of input image
8 Results

To test the quality of noise reduction, the PSNR (Peak Signal-to-Noise Ratio) metric is usually used. But this metric is incorrect for datasets where there is no clean samples. PSNR between output of the architecture which not completely reduced noise and average series value can be more then PSNR between output of the architecture which completely reduce noise component and the average series. This is due to the fact that a noise component remains in the average value of the series. In this case PSNR measure incorrect, but we have not another way to comparison our solution with state-of-the-art denoising approaches.

We compare our method with state-of-the-art approaches for training unsupervised image denoising neural networks. Before comparison neural network of each work we train by authors pipeline on our dataset. In addition for comparison taken an unsupervised image denoising algorithm from the OpenCV library: Non-local means denoising [11]. Table 1 gives comparison results. Furthermore visual comparison of inference all approaches is shown (see Fig. 7). Additional shown comparison with Non-local means denoising [11] (see Fig. 8).

| Approach            | PSNR |
|---------------------|------|
| Noise2Noise         | 34.6 |
| Noise2Void          | 30.53|
| Noise2Self          | 33.7 |
| Non-local means denoising | 36.04|
| Deep image prior    | 36.34|
| Our                 | **37.1** |

Table 1. Comparison proposal approach with related works

These models work only with frames received from the CMOS sensor from which the dataset was obtained for their training. An example of work on frames from another sensor is shown (see Fig. 9). Poor prediction quality is noticeably.

9 Summary

As a result pipeline was implemented for training the noise reduction model. It does not depend on datasets with clean sample images. A neural network architecture was implemented and tested to effectively solve the noise reduction problem without clean data.
Fig. 7. From left to right on first line: image obtained by averaging all frames in series, original image, result of Noise2Noise, result of Noise2Self, from left to right of second line: result of Noise2Void, result of the non-local means denoising, result of Deep Image Prior, result of the architecture with frequency decomposition of input signal.

Fig. 8. From left to right: original image, image obtained by averaging all frames in series, result of the architecture with frequency decomposition of input signal, result of the non-local means denoising.

According to the results, proposed model solves the optimization problem (3) quite efficiently. All experiments were performed on the dataset collected for this work.

The source code is available at the following link: https://github.com/AlexeySr us/image_denoising.
Fig. 9. From left to right: original image, image obtained by averaging all frames in series, result of the architecture with frequency decomposition of input signal, result of the non-local means denoising

References

1. Lehtinen, J., Munkberg, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2Noise: learning image restoration without clean data. CoRR abs/1803.04189 (2018). http://arxiv.org/abs/1803.04189
2. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.: Unprocessing images for learned raw denoising. CoRR abs/1811.11127 (2018). http://arxiv.org/abs/1811.11127
3. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. CoRR abs/1707.01313 (2017). http://arxiv.org/abs/1707.01313
4. Krull, A., Buchholz, T.-O., Jug, F.: Noise2Void - learning denoising from single noisy images. CoRR abs/1811.10980 (2018). http://arxiv.org/abs/1811.10980
5. Batson, J., Royer, L.: Noise2Self: blind denoising by self-supervision. CoRR abs/1901.11365 (2019). https://arxiv.org/abs/1901.11365
6. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. CoRR abs/1711.10925 (2017). https://arxiv.org/abs/1711.10925
7. GSMARENA. https://www.gsmarena.com/apple_iphone_x-8858.php. Accessed 20 Jun 2020
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385
9. Liyuan, L., et al.: On the variance of the adaptive learning rate and beyond. CoRR abs/1908.03265 (2019). http://arxiv.org/abs/1908.03265
10. Ayhan, M., Berens, P.: Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. In: MIDL 2018 Conference (2018). https://openreview.net/forum?id=rJZz-knjz
11. Buades, A., Coll, B., Morel, J.-M.: Median filtering: a new insight. Image Process. On Line (2011). https://doi.org/10.5201/ipol.2011 bcm_nlm