Convolutional neural network for noise reduction to improve the quality of images obtained using active-pulse television measuring systems

A Novgorodov, N Borodina and M Kuryachiy
Tomsk State University of Control Systems and Radioelectronics, 40, prospect Lenina, 634050, Tomsk, Russia
E-mail: shaman.king.481516@gmail.com

Abstract. The paper presents a software implementation of an image recovery algorithm using a convolutional neural network. An example of using the developed algorithm in the context of noise reduction of images obtained using active-pulse television measuring systems is presented. The result obtained with a convolutional neural network is better than classical filtering algorithms (PSNR higher by 2.01 dB).

1. Introduction
Technical vision systems are widespread in science and technology. These systems are used to solve problems of measurement, search and recognition of objects of interest. The performance of such systems is significantly degraded by difficult viewing conditions such as fog, smoke, dust, rain, etc. The operating range of television systems depends on back propagation interference arising from the scattering of light photons in atmospheric aerosols. There are television systems that are capable of operating effectively in challenging vision environments. One of such classes is active-pulse television measuring systems (AP TMS).

The AP TMS feature is the use of pulsed illumination, synchronized with the pulse exposure of the photodetector shutter. The pulsed operating mode of the system allows you to form a vision zone at a certain distance. By controlling the observation range, you can create a "depth map" of space [1].

The AP TMS generates images that contain various noise and interference. Noise and interference in digital images can be divided into three components: additive, multiplicative, applicative.

The aim of the presented work is to implement a model of a convolutional neural network for noise reduction and to study its effectiveness in recovering images obtained using the AP TMS.

2. Denoising Convolutional Neural Network
Currently, in addition to classical noise reduction algorithms, algorithms based on machine learning are widely used, in particular, algorithms based on convolutional neural networks.

An artificial neural network is a type of machine learning that has been substantiated by the principles of the nerve cells of a living organism. An artificial neural network is a sequence of neurons connected by synapses. A neuron is a computational unit that receives information, performs simple calculations on it and transfers the result further [2].

The output value of a neuron for N input values is determined by the formula:

\[ y = \sum_{i=0}^{N} f(W_i x) \] (1)
This formula uses a function called the activation function. Activation functions can perform different tasks depending on their type. Some activation functions are designed to normalize input or output values, and some are designed to add non-linearity to the network. This work uses the ReLU activation function. The ReLU activation function is defined by the following formula:

\[
ReLU(x) = \begin{cases} 
0, & x < 0 \\
 x, & x \geq 0
\end{cases}
\]

A sample from the training set is fed to the input of the model, and the result of the algorithm is compared with the expected answer. Then the coefficients of the algorithm are changed in such a way that the result of the algorithm's work matches the expected answer.

Convolutional neural networks are widely used to solve computer vision problems. In a conventional neural network, each neuron is connected to all the neurons of the previous layer, and each connection has its own weighting factor. This neural network architecture does not take into account the structure and relative position of the data.

To solve the problem of improving the quality of images obtained with the AP TMS, the PyTorch library was used, in which the architecture of a convolutional neural network was built and trained. The neural network consists of an input convolutional layer and a given number of identical hidden layers [3]. The ReLU function was selected as the activation function at all stages. All hidden layers have a batch normalization operation. A block diagram of the neural network is shown in Figure 1.

![Figure 1. Noise Reduction Convolutional Neural Network Architecture.](image)

The BSD300 (Berkeley Segmentation Dataset) dataset from the University of Berkeley was used to train the neural network model. The dataset was designed for segmentation tasks and contains 300 reference images of various resolutions [4].

To train the model, a trainer and test dataset loader was implemented. The loader returns two images: distorted with normal noise and the original "without noise". A complete pass of the neural network over the training data is called the training period. The learning algorithm consists of a given number of learning periods. Within one period, both the training of the network and its validation take place [5]. Validation checks the neural network model against a certain amount of test data to prevent overfitting.
The «Adam» algorithm [6] was chosen as the neural network optimizer. The block diagram of the neural network learning algorithm is shown in Figure 2.

![Neural network learning algorithm](image)

**Figure 2.** Neural network learning algorithm.

The training of the neural network model took place over 50 periods. Plots of loss function versus learning period for training (solid line) and validation data (dashed line) are shown in Figure 3.

![Dependence of the loss function on the training epoch](image)

**Figure 3.** Dependences of the loss function on the training epoch.
The lack of a reference makes it difficult to assess image quality. For test images, it is possible to evaluate noise on uniform white and black areas of the image. In this case, the average value of the uniform area is selected as a standard. An experimental image with AP TMS is shown in Figure 4.

There is a clear linear dependence in the experimental images, and therefore, the value of linear polynomial regression can be chosen as a reference. To evaluate the noise, we select a uniform area on the banner with the number "12". The experimental data obtained with the AP TMS and their polynomial regression, are presented in Figure 5.

3. Analysis of results
To compare the results of image restoration, we used classical noise reduction algorithms, such as the Gaussian filter and the median filter. A Gaussian filter is a linear spatial filter that uses a normal distribution for the averaging "mask" coefficients. The median filter uses a sorted sample to obtain the
median value. The dimensions of the Gaussian filter and the median filter are 3x3 elements. The standard deviation of the Gaussian filter is 3 elements on both axes.

The standard deviation and the peak signal-to-noise ratio are used to assess the image quality [7]. The standard deviation between images I and K is calculated by the following formula:

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

(2)

\[ m, n \] – width and height of images respectively

The peak signal-to-noise ratio is calculated using the following formula:

\[ PSNR = 10 \cdot \log \left( \frac{\text{MAX}_I^2}{MSE} \right) \]  

(3)

\[ \text{MAX}_I \] – maximum possible pixel value

Table 1 presents the results of evaluating the image quality relative to polynomial regression for various reconstruction algorithms.

| Table 1. Quality assessment results relative to polynomial regression of experimental data |
|-------------------------------------------|----------|--------|----------|
| **Mean value**               | **MSE**  | **PSNR, dB** |
| Experimental data            | 194.10   | 18.59  | 35.43    |
| Gaussian filter              | 194.12   | 13     | 36.98    |
| Median Filter                | 194.03   | 13.98  | 36.67    |
| Neural network               | 194.11   | 11.72  | 37.44    |

From Table 1 it can be seen that the result obtained with the convolutional neural network is better than the classical filtering algorithms (PSNR is 2.01 dB higher). The study shows that a neural network can improve performance when using a larger training set. Forming your own data set, taking into account the specifics of AP TIS, can also improve the performance of the neural network.

Acknowledgments

The research was carried out at the expense of the Russian Science Foundation grant No. 21-79-10200 at TUSUR.

References

[1] Kapustin V V, Movchan A K and Kuryachiy M I 2017 Vision Area Parameters Analysis for Active-Pulse Television-Computing Systems. *Int. Siberian Conf. on Control and Communications (SIBCON)*, 2017 1–4

[2] Nikolenko S, Kadurin A and Arkhangelskaya E 2020 Deep Learning. Immersion in the world of neural networks (St. Petersburg: Piter) p 480

[3] Shukla N 2018 Machine Learning with TensorFlow (Shelter Island: Manning Publications) p 272

[4] Paszke A et al. 2019 PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 32 8024–8035

[5] Zhang K, Zuo W, Chen Y, Meng D and Zhang L 2017 Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, *IEEE Transactions on Image Processing*. 26 7 3142-3155

[6] Kingma D P, Ba J L 2015 Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*, ICLR 2015.

[7] Wang Z and Bovik A C 2009 Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures, *IEEE Signal Processing Magazine*. 26 1 98-117