ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

Experimental Analysis of Image Denoising using Convolution Neural Network Based on MATLAB

Rajni
ECE Department, DCRUST, Murthal, India

Pawan Kumar Dahiya
ECE Department, DCRUST, Murthal, India

***Abstract* - The image enhancement method can help in to amplify the picture quality. Pull out the noise from the original representation of an image is a big challenge in today's life. Deep CNN has delighted significant attention in the domain of image de-noising. Deep learning plays a significant role. Deep learning got great success in the field of image enhancement, such as low-level convolution neural networks (LLCNN), high-level convolution neural networks (HLCNN), super-resolution (SRCNN), and de-noising convolution neural networks (DnCNN). In this article author using the Deep CNN technique for image de-noising. MATLAB tool is used for overall research. In this paper, the static noise level has been taken, and carried out the different experimental designs in the parameters such as PSNR and SSIM for de-noising the image. In this, add a salt and pepper noise at a specific noise level. This article proposes a convolution neural network based on the deep network technique. An approach that has been found in this paper is received a new pixel patch value using CNN with different parameters. The table also differentiates the parameters by adding salt and pepper noise at different noise levels. It can also be applied to the real-time world..**

Index Terms - Deep Convolution Neural Network, Image de-noising, PSNR, SSIM

INTRODUCTION

Image De-noising is the task of removing noise from an image. It is an extraction of a signal by adding a mixture of signal and noise. Image de-noising aims to reclaim a clear picture from a noisy picture. It is a outstanding-inverse issues in image processing. Image de-noising methods can improve the original image quality and bring back all features that lie in an image or an original signal; they are acceptable in different areas, such as biomedical imaging, remote-sensing image, etc. For a rowdy image y, we can say that rowdy as noisy image, the issue in image de-noising, that can be showed by $x + \upsilon = y$, where the actual image is x (also mentioned to as the polished picture or signal), and represents additive Gaussian noise is υ (AWGN) with σ standard deviation. [1] Fig. 1 is showing here the trained de-noising network with noisy images and the de-noise image. Here is showing a trained de-noising network. Intelligent retrieval is a branch of machine learning used in artificial intelligence; intelligent retrieval, also known as deep learning, has multiple networks capable of different types of learning, supervised and unsupervised. It is Also known as deep neural learning or deep neural network. [2]

Fig.1 Trained decision network

Classification of Image de-noising are various types mains are as follows:

- Spatial domain: spatial domains are also two types linear and nonlinear.
- Transform domain: These are also two types data-adaptive and data non-adaptive.

Image de-noising using CNN also has different types like in linear we can classify it as mean and wiener filter in linear. Nonlinear has a median and weighted median. Like this transform divided into adaptive and non-adaptive, adaptive further divided as wavelet and spatial frequency. [3]

In deep learning used the batch normalization term plays an important role in deep learning. It is a technique for training very deep neural networks that define the inputs to each sub-batch layer. This has the effect of balancing the learning process. It is also used for reducing the total number of training time required to train deep networks. Batch normalization works by subtracting an input layer into the sub-batch mean. It can be splitting by the small-batch standard deviation. Batch normalization connect two trained variable, gamma γ and beta β, which can scale and shift the normalized value.[4] Image de-noising is the process when we want to reduce or the unwanted part from the input signal or picture**.** A deep network convolution neural network is best for image de-noising. It performs a deep network layer for its training process. It is repeated at every layer during the training time. A convolution neural network is best for solving the problem of noise reduction, super-resolution.[5]

The architecture of machine and deep networks learning processes and deep learning processes is shown in fig. 2; shows here that, during the machine learning process, an example taking a car image as input, feature extraction has been performed separately, and then apply convolution neural network to classify the different image that it is a car image. So in the machine learning process of feature extraction and classification are done separately. While we talk about the deep learning feature extraction and classification done together or we can say that both processes done during the convolution multilayer [6]

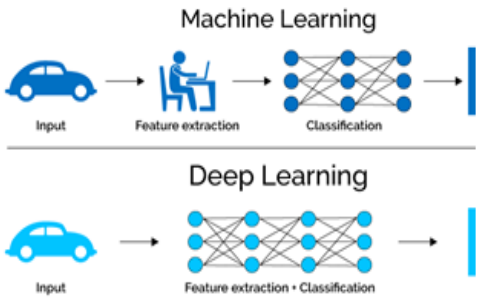


Fig.2 Architecture Deep CNN (Machine vs Deep learning)

Fig. 3 is showing here the architecture of the convolution neural network for image de-noising. A noisy image is taken as input in this architecture, then apply an edge detection technique to find its edges. After getting edges, make a convolution network that has batch normalization and the Relu layer. Batch normalization is a technique for training very deep neural networks that define the inputs to each sub-batch layer. This has the effect of balancing the learning process. It is also used for reducing the total training time required to train deep networks. Batch normalization works by subtracting an input layer into the mean of a sub-batch. It can be splitting by the sub-batch standard deviation. Batch normalization includes two trainable parameters, gamma γ and beta β, which can scale and shift the normalized value. Relu layer is used as the function f(x) = max(0, x) to all of the values in the input. We can say that this layer is used for the changes in all the negative activations to 0, and this layer also increases the properties of the nonlinear network. After applying for an image, pass through a fully connected flatten layer and get de noise output image.

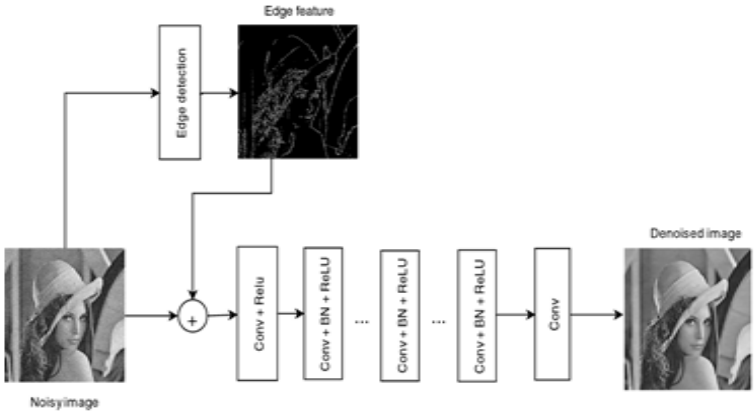


Fig. 2: CNN network for Image De-noising

METHODOLOGY

The methodology is showing as workflow in fig. 4. First, we have to take an input image and then add salt and pepper noise. This is a noise that has sometimes been seen in images. It is also known as impulse noise. The sharpness can produce this type of noise in an image or unexpected disturbances in the image. We can remove this type of noise using a median filter or a morphological process. Batch normalization is a technique for training very deep neural networks that define the inputs to each sub-batch layer. This has the effect of balancing the learning process. It is also used for reducing the total training time required to train deep networks.[4]

Deep convolution neural network, after adding noise apply DnCNN. This layered network shows convolution and Relu layer; this is also known as rectified linear unit, it is not a different component of the convolutional neural network. At the same time, it is an additional step for convolution operation. The main motive of applying the rectifier function is to boost up the non-linearity feature in the input images. Then image passes through the next layer, a combination of convolution, Relu, and batch normalization. Then it passes through the fully connected layer that is convolution. It is a feed-forward network. Fully Connected layers in neural networks play a very serious or great role in the deep network; these are the layers. All input from each layer is the activation function for the next layer. In the machine learning model, it is well-liked. In this, some layers are fully connected layers that compile the facts take out by previous. Workflow for CNN is also given in fig. 4, which describes how our DnCNN is worked out. We have to take input image data at the first step, then add noise to it. After adding noise, we have to apply our proposed technique, deep CNN, to increase the image quality and remove the noise from the original image. We are also stressing to evaluate the parameters like PSNR and SSIM. [7]

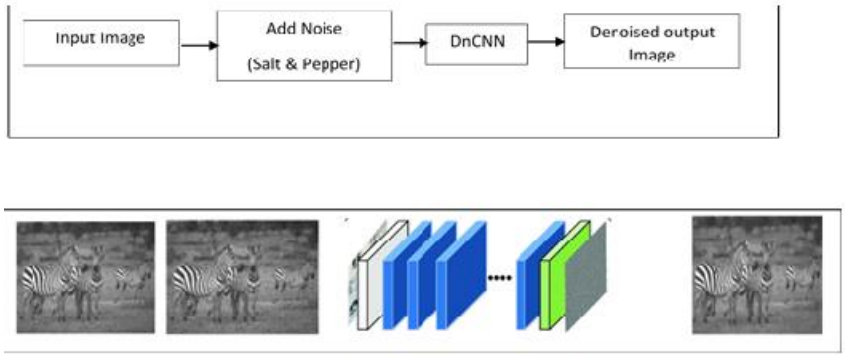


Fig. 4 Architecture / workflow of DnCNN

In this paper, MATLAB has been used for deep network image denoising. The algorithm is also given with all suitable steps. Basic code for reading a denoise DnCNN image is given below:

```
net = denoising network('DnCNN'); ...

I = imread('pepper.png')

montage({I,noisyI}) title('Original Image (Left) and
Noisy Image (Right)')

denoisedI = denoiseImage(noisyI,net);
imshow(denoisedI) title('Denoised Image')
```

Deep learning as deep networks, recurrent networks, machine networks, and different artificial networks are used for deep learning. The human brain inspires these networks. [8][9].

We have to use the MATLAB tool and used the convolution neural network toolbox; we can also continue this in GUI based. For this deep learning process, we can also use another tool such as python.

Workflow for CNN is also given in fig. 4, which describes how our DnCNN is worked out. We have to take input image data at the first step, then add noise to it. After adding noise, we have to apply our proposed technique, deep CNN, to increase the image quality and remove the noise from the original image. We are also stressing to evaluate the parameters like PSNR and SSIM.

Fig. 3 is showing here the architecture of the convolution neural network for image denoising. A noisy image is taken as input in this architecture, then apply an edge detection technique to find its edges.

The basic algorithm is describing as below with description steps. An input image is taken, and calculate their different parameters to get a new pixel value. The brief description is as below:

RESULT & DISCUSSION

Algorithm Pseudocode of DnCNN for image denoising
For all the pixels $A_{ij} \epsilon J$ **do**
If $A_{ij} \notin \{0,1\}$ ***then***
retain A_{ij}
Continue;
While $A_{ij} \epsilon \{0,1\}$ **do**
Initialize R = 1;
Compute H_{ij}^{R}
Compute $\rho_{P_{ij}^{R}}$, φ_{ij}^{R}
Compute F_n , $\rho_{P_{ij}^{R}}$

$$F_n = \wedge(\vee(\triangle_{ij}))$$
$$\rho_{P_{ij}^{R}} = \frac{\sum_{k=1}^{m} \triangle_{ij}}{m} \quad \forall\, i = 1 \ldots \ldots .. M$$

if $\rho_{P_{ij}^{R}}(A_{ij}) \geq F_n$ **then**
retain A_{ij}
brake;
if $\varphi_{ij}^{R} \leq \epsilon$ **then**

$$A_{ij} = n_{avg}$$
$$n_{avg} = \frac{\sum_{k=1}^{m} N_k}{m}$$

brake;
Compute K_{ij}^{R}

$$\mu = \left| K_{ij}^{R} \right|$$

if $\mu < 1$ **then**
R = R +1;
continue;

Steps Description of Algorithm

STEP1: J is the image, A_{ij} Is each pixel at location (ij) and i is represent the row value, and j is the column of the matrix.
STEP2: The noise intensity values are either 0 or 1. So each pixels value $A_{ij} \notin \{0,1\}$ is retain.
STEP3: The H_{ij}^{R} Is neighborhood pixel set with a window of size R.

$$H_{ij}^{R} = \left\{A_{i+k,\ j+l} \forall k, l \epsilon [-R, R]\right\}$$

STEP4: In this step, Gaussian membership functions ($\rho_{P_{ij}^{R}}$) and Variance (φ_{ij}^{R}) is calculated.

$$\rho_{P_{ij}^{R}} = \frac{\sum_{k=1}^{m} \triangle_{ij}}{m} \quad \forall\, i = 1 \ldots \ldots .. M$$

STEP5: In this step F_n , $\rho_{P_{ij}^{R}}$ Is Computed.

$F_n = \wedge(\vee(\triangle_{ij}))$ whereas $\wedge$,$\vee$ are min and max operators,

$$\triangle_{ij} = \begin{bmatrix} \rho_{P_{ij}^{(R,1)}}(r_1) & \rho_{P_{ij}^{(R,1)}}(r_2) \cdots & \rho_{P_{ij}^{(R,1)}}(r_n) \\ \rho_{P_{ij}^{(R,2)}}(r_1) & \rho_{P_{ij}^{(R,2)}}(r_2) \cdots & \rho_{P_{ij}^{(R,2)}}(r_n) \\ \vdots & \vdots \quad \ddots & \vdots \\ \rho_{P_{ij}^{(R,h)}}(r_1) & \rho_{P_{ij}^{(R,h)}}(r_2) \cdots & \rho_{P_{ij}^{(R,h)}}(r_n) \end{bmatrix}$$

STEP6: In this step, the condition is checked.

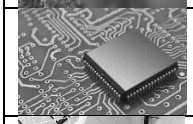
if $\rho_{P_{ij}^R}(A_{ij}) \geq F_n$ **then**
retain A_{ij}
brake;
STEP7: In this step, check the condition
if $\varphi_{ij}^R \leq \epsilon$ **then**

$$A_{ij} = n_{avg}$$
$$n_{avg} = \frac{\sum_{k=1}^{m} N_k}{m}$$

brake;
STEP8: In this step K_{ij}^R Is calculated. In high-level noise, μ becomes zero; then, the R-value is increased by one, and the process is repeated again. [8][10]
if $\mu < 1$ **then**
R = R +1;
continue;
STEP9: This is the last step where a new pixel's (A_{ij}^{new}) value is obtained

$$A_{ij}^{new} = \frac{K_{ij}^R}{w} \; ; \; w$$

Table. 1 Evaluation of SSIM and PSNR (dB) at a different noise level of the image de-noising using CNN method on different used testing images.

	SSIM value at different Noise Level using CNN					**PSNR value at different Noise Level using CNN**				
Input Image	**5% (0.05)**	**10% (0.10)**	**15% (0.15)**	**20% (0.20)**	**25% (0.25)**	**5% (0.05)**	**10% (0.10)**	**15% (0.15)**	**20% (0.20)**	**25% (0.25)**
	0.9955	0.9910	0.9876	0.9826	0.9768	45	43	39	36	32
	0.9957	0.9915	0.9869	0.9819	0.9753	43	38	34	32	28
	0.9956	0.9942	0.9855	0.9786	0.9703	42	37	35	32	29
	0.9928	0.9869	0.9802	0.9734	0.9641	46	44	40	37	32
	0.9928	0.9859	0.9786	0.9710	0.9610	45	42	38	34	30

In this section table.1 is showing the value of different parameters of image de-noising at a different level of noise like SSIM and PSNR. Noise level is taking at 5 to 25%. Between these, noise level structural Similarity index and peak signal to noise ratio have been calculated. There are taking five different input images. This is showing the parameters evaluated after adding the salt and pepper noise. Fig. 6 is showing the graph of the different PSNR values at different noise levels for various images. It is showing at 0.05 noise level, PSNR is 45 of Lina image when increasing the noise level at 0.15 then PSNR decreases to 38, or at 0.25 noise level it is 30, so when increasing the noise level, it decreases the PSNR values. Fig. 5 is showing the graph of SSIM values that are obtained at different noise levels. As shown in fig; at the 0.05 noise level SSIM

value is 0.9928; on 0.15 noise level, it is 0.9786, and 0.25 it is 0.9610. So it is concluded at higher noise levels, SSIM decreases.[11][12].

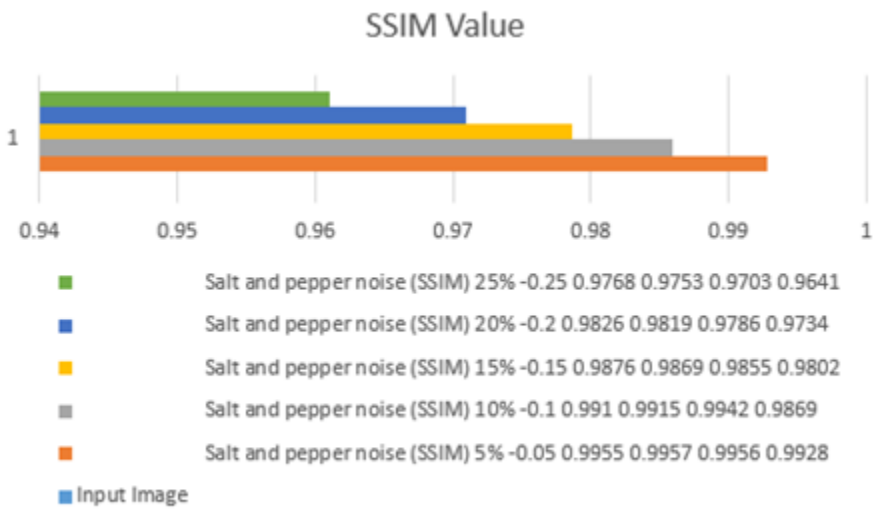


Fig. 5 Performance Evaluation values of SSIM at different noise level

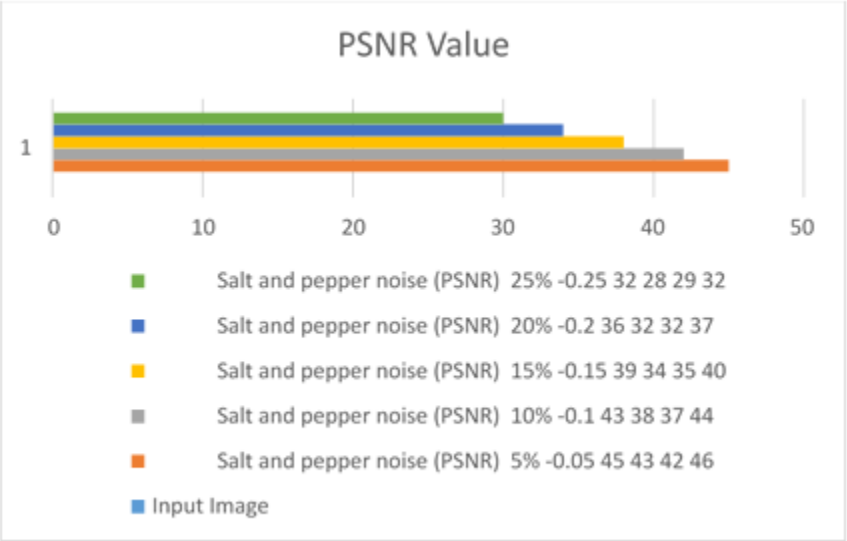


Fig. 6 Performance Evaluation values of PSNR at different noise level

CONCLUSION AND FUTURE SCOPE

In this article, a deep convolution neural network was brought forward for denoising of images. This paper has taken noise at different noise intensity levels with a Gaussian membership function and its variance. On these noise levels found different parameters like PSNR and SSIM with the help of MATLAB software. An approach that has been found in this paper is to receive a new pixel patch value using CNN with different parameters. This can also train an image data set for the classification of images to identify an image. We will stress on the investigation another proper deep CNN network for real-time problems because real-time systems are very complex than traditional and evaluate other parameters like execution time, MSE and BER, etc.

REFERENCES

[1]. K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017, doi: 10.1109/TIP.2017.2662206.

[2]. A. Kaur and G. Kaur, "A review on image enhancement with deep learning approach," *Accent. Trans. Image Process. Comput. Vis.*, vol. 4, no. 11, pp. 16–20, 2018, doi: 10.19101/tipcv.2018.411002.

[3]. R. Ahmadi, J. Kangarani Farahani, F. Sotudeh, A. Zhaleh, and S. Garshasbi, "Survey of image denoising techniques," *Life Sci. J.*, vol. 10, no. 1, pp. 753–755, 2013, doi: 10.5120/9288-3488.

[4]. C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020, doi: 10.1016/j.neunet.2019.08.022.

[5]. M. Elhoseny and K. Shankar, "Optimal bilateral filter and Convolutional Neural Network based denoising method of medical image measurements," *Meas. J. Int. Meas. Confed.*, vol. 143, pp. 125–135, 2019, doi: 10.1016/j.measurement.2019.04.072.

[6]. X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, 2017, doi: 10.1109/TIP.2017.2691802.

[7]. B. Liu and J. Liu, "Overview of Image Denoising Based on Deep Learning," *J. Phys. Conf. Ser.*, vol. 1176, no. 2, 2019, doi: 10.1088/1742-6596/1176/2/022010.

[8]. S. Ahmad Radzi, "A MATLAB-based Convolutional Neural Network Approach for Face Recognition System," *J. Bioinforma. Proteomics Rev.*, vol. 2, no. 1, pp. 1–5, 2016, doi: 10.15436/2381-0793.16.009.

[9]. Y. Quan, Y. Chen, Y. Shao, H. Teng, Y. Xu, and H. Ji, "Image denoising using complex-valued deep CNN," *Pattern Recognit.*, vol. 111, p. 107639, 2021, doi: 10.1016/j.patcog.2020.107639.

[10]. J. Latif, C. Xiao, A. Imran, and S. Tu, "Medical imaging using machine learning and deep learning algorithms: A review," *2019 2nd Int. Conf. Comput. Math. Eng. Technol. iCoMET 2019*, pp. 1–5, 2019, doi: 10.1109/ICOMET.2019.8673502.

[11]. M. K. Mihçak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Process. Lett.*, vol. 6, no. 12, pp. 300–303, 1999, doi: 10.1109/97.803428.

[12]. T. Saba, A. Rehman, and G. Sulong, "An intelligent approach to image denoising," *J. Theor. Appl. Inf. Technol.*, vol. 17, no. 1, pp. 32–36, 2010.

[13]. D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, "Insect detection and classification based on an improved convolutional neural network," *Sensors (Switzerland)*, vol. 18, no. 12, pp. 1–12, 2018, doi: 10.3390/s18124169.

[14]. C. Tang, Q. Zhu, W. Wu, W. Huang, C. Hong, and X. Niu, "PLANET: Improved Convolutional Neural Networks with Image Enhancement for Image Classification," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/1245924.

[15]. L. Tao, C. Zhu, G. Xiang, Y. Li, H. Jia, and X. Xie, "LLCNN: A convolutional neural network for low-light image enhancement," *2017 IEEE Vis. Commun. Image Process. VCIP 2017*, vol. 2018-Janua, no. 2013, pp. 1–4, 2018, doi: 10.1109/VCIP.2017.8305143.

[16]. Y. Li, H. Wang, L. M. Dang, A. Sadeghi-Niaraki, and H. Moon, "Crop pest recognition in natural scenes using convolutional neural networks," *Comput. Electron. Agric.*, vol. 169, no. December 2019, p. 105174, 2020, doi: 10.1016/j.compag.2019.105174.

[17]. L. Hu, M. Qin, F. Zhang, Z. Du, and R. Liu, "RSCNN: A cnn-based method to enhance low-light remote-sensing images," *Remote Sens.*, vol. 13, no. 1, pp. 1–13, 2021, doi: 10.3390/rs13010062.

[18]. W. Wang, X. Wu, X. Yuan, S. Member, and Z. Gao, "An Experiment-Based Review of Low-Light Image Enhancement Methods," vol. 8, 2020, doi: 10.1109/ACCESS.2020.2992749.

[19]. S. Anwar, C. Li, and F. Porikli, "Deep Underwater Image Enhancement," pp. 1–12, 2018, [Online]. Available: http://arxiv.org/abs/1807.03528.

[20]. K. Singh, A. Seth, H. S. Sandhu, and K. Samdani, "A comprehensive review of convolutional neural network based image enhancement techniques," *2019 IEEE Int. Conf. Syst. Comput. Autom. Networking, ICSCAN 2019*, pp. 1–6, 2019, doi: 10.1109/ICSCAN.2019.8878706.

[21]. T. Kaur and R. K. Sidhu, "Performance Evaluation of Fuzzy and Histogram Based Color Image Enhancement," *Procedia Comput. Sci.*, vol. 58, pp. 470–477, 2015, doi: 10.1016/j.procs.2015.08.009.