ISSN 2712-0554
Original Research

Heritage and Sustainable Development
Vol. 2, No. 2, July 2021, pp.89-96
https://doi.org/10.37868/hsd.v2i2.71

Facial image denoising using AutoEncoder and UNET

Milan Tripathi[1*]
[1]Department of Computer Engineering, Tribhuvan University, Lalitpur, Nepal

*Corresponding author: milan.tripathi@acem.edu.np

Abstract

Image denoising is a crucial topic in image processing. Noisy images are generated due to technical and environmental errors. Therefore, it is reasonable to consider image denoising an important topic to study, as it also helps to resolve other image processing issues. However, the challenge is that the classical techniques used are time-consuming and not flexible enough. This article compares two major neural network architectures which look promising to resolve these issues. The AutoEncoder and UNET is now the most researched subject in deep learning for image denoising. Multiple model architectures are designed, implement, and evaluated. The dataset is preprocessed and then it is used to train and test the model. It is clearly shown in this paper which model performs the best in this task by comparing both models using the most used parameters to evaluate image quality PSNR and SSIM.

Keywords: AutoEncoder, UNET, Peak signal-to-noise ratio (PSNR), Structural Similarity (SSIM)

1. Introduction

Image denoising is a hot topic in deep learning and image processing. It is a basic problem in the field of image processing and computer vision, with the underlying purpose of estimating the original image by removing noise from a noise-contaminated representation of the image. Due to environmental and technical error often noised image gets generated. As a result, image denoising is important in a variety of applications, including image reconstruction, visual monitoring, image registration, image segmentation, and image classification, where retrieving the original image material is critical for strong results. Although several algorithms have been proposed for denoising the image, the issue of image noise suppression remains an open challenge, especially when the images are captured under poor conditions with a high noise level. Different classical techniques like Spatial domain filtering, variational denoising methods [1] were used in the past. With the advancement in Deep learning, many researchers are trying to resolve the problem by using Neural Networks.

Image denoising is a crucial task in image processing and deep learning. Different classical techniques and modern development are explained in his paper [1]. Different classical techniques like Spatial domain filtering, Transform Domain filtering, and modern techniques like CNN-based denoising methods are discussed is explained. Ali Awad [6] purposed a method to remove the noise from an image corrupted from impulse, gaussian, or a mixture of both. The method is based on divided into two, in which the first is removing the small noise component, and subsequent steps are based on principal component analysis. The process is assumed to remove the majority of noise in the first stage and smaller ones later. Olaf Ronneberger [2] purposed a UNET structure for the first time for the segmentation of biomedical images. In this paper, a U-shaped model was introduced where unlike old models a skip connected between encoding and decoding layer was introduced which allows some data to flow and help in better image generation. Irfan Ali [3] purposed an AutoEncoder model for image denoising with Color Scheme. The work investigates performing denoising on the RGB dataset. Gaussian noise of 0.2 factor is added in all the images of the dataset and an autoencoder is used to remove the noise. Latha H N [4] purposed a local modified UNET Architecture for Image Denoising. The work investigates the UNET model [2] in removing the noise and compares it with the local modified UNET Architecture. The model is trained in three types of noises Gaussian, Salt&Pepper, and Camera Shake. The model can gain an optimal PSNR and SSIM value.

S. Ghose [7] purposed a CNN model to remove noise from an image and restore it to a high-quality image. The analysis is done only for Gaussian noise for different percentage Gaussian white noise and comparison traditional method is also done. O. Sheremet [8] purposed a CNN-based model for denoising images in Info communication systems. J. Gurrola-Ramos, [9] purpose a dense U-Net neural network. To denoise the picture, a residual Dense U-Net Neural Network was used. The intended model has several characteristics, such as the fact that the denoising mechanism does not require prior knowledge of noise.

Chunwei Tian [10] purposed a new type of network called batch-renormalization denoising network (BRDNet) which combines two networks to expand the network's width and thereby gain additional capabilities. The model outperforms state-of-the-art models in terms of efficiency.

All the research papers provide information about various processes, but none of them provide a comparison with the AutoEncoder and UNET. Thus, this research paper fills that gap and provides a reason for choosing the model with optimal performance. In this paper, I would like to design, implement and compare a basic AutoEncoder [14] model and UNET for image denoising. Since the classical techniques being used are not flexible and different are used for different noises which consume more time and effort, a neural network-based model can learn fast and provide better results. Comparing the result of the models will give a better idea about the performance of the model which is not done by previous researchers. This research will provide a better indication for use of the proper model when performing the image denoising.

In this paper, a basic AutoEncoder and a modified UNET architecture are designed and implemented for image denoising purposes. The modification is done as per the requirement for experimenting. The FER2013 dataset [5] is used for training and testing the model. Three kinds of noise will be added to the image, Gaussian, Poisson, and Salt & Pepper noise which generate noised data for training and testing the model. The noise is added in a fixed noise ratio value to compare the performance of both models in the same constraint. Proper analysis using the model used metrics PSNR [13] and SSIM [12] is done and the visualization of the image for each operation is also done. PSNR was used before SSIM, it is simple, has been commonly used in various digital image measurements, and it is considered validated and valid. SSIM is a newer measuring instrument that is intended to best accommodate the functioning of the human sensory system by focusing on three factors: luminance, contrast, and form.

2. Deep learning model

Initially, a basic AutoEncoder model is designed and implemented. The objective of the experiment is to produce a higher quality noise-free image, but the model is unable to do so. Even increasing the model size results in a decrease in the n efficacy of generating the images. As the model lost more information when the large model is used, UNET model architecture is used. The UNET model consists of skip connections that pass certain information from the encoding part to the decoding part Thus, two model architecture are considered which are Basic Autoencoder Architecture and UNET Architecture.

2.1. Basic AutoEncoder architecture

Autoencoders are the type of neural network that is trained in an unsupervised manner to learn the compressed representation of raw data. Fig. 1 is the basic representation of the autoencoder. An autoencoder consists of three parts Encoder, bottleneck, and decoder. The encoder compresses the image into a lower dimension and the decoder reconstructs the image from this representation. The compressed dimension is also known as the bottleneck is the lowest possible of the input data. The architecture formation is lost in this process, but the most important gets preserved. Autoencoder is widely used for image processing tasks like image deblurring, denoising, compression, etc. The architecture of the Autoencoder used for the experiment is shown in Fig. 2.

This Autoencoder can generate noise-free images but there is certain blurriness in the image which can be easily seen in Fig. 7-9. To alleviate this issue, large AutoEncoders are designed and implemented, but none can resolve the problem.

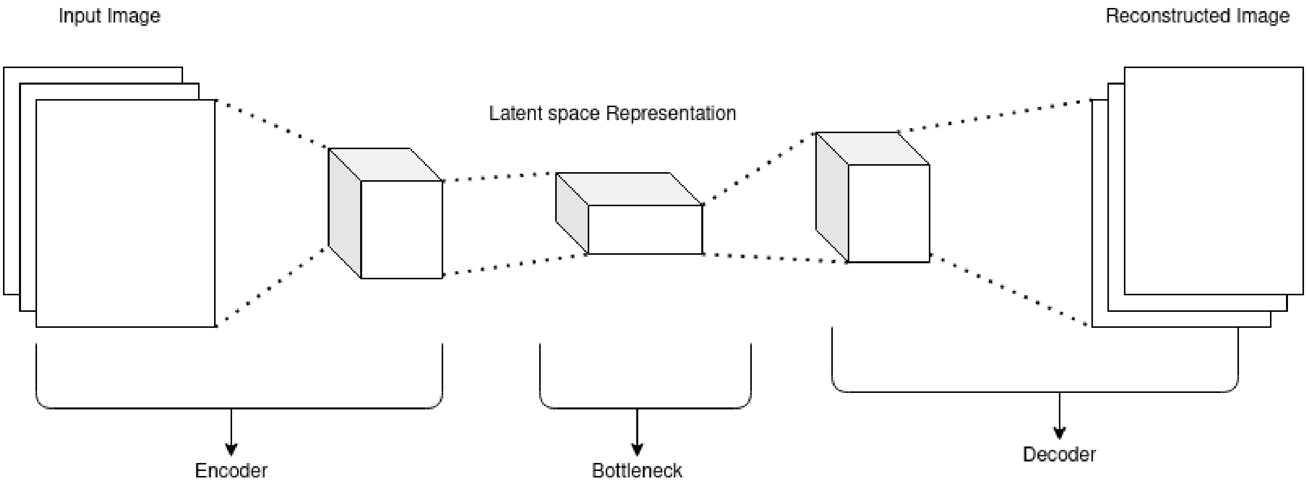
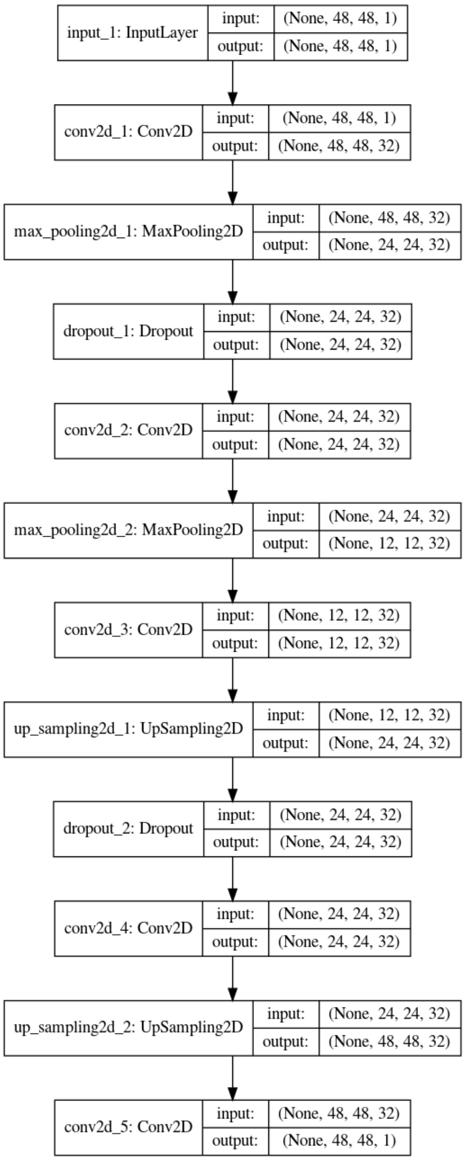


Figure 1. Basic architecture AutoEncoder



input_1: InputLayer	input:	(None, 48, 48, 1)
	output:	(None, 48, 48, 1)
conv2d_1: Conv2D	input:	(None, 48, 48, 1)
	output:	(None, 48, 48, 32)
max_pooling2d_1: MaxPooling2D	input:	(None, 48, 48, 32)
	output:	(None, 24, 24, 32)
dropout_1: Dropout	input:	(None, 24, 24, 32)
	output:	(None, 24, 24, 32)
conv2d_2: Conv2D	input:	(None, 24, 24, 32)
	output:	(None, 24, 24, 32)
max_pooling2d_2: MaxPooling2D	input:	(None, 24, 24, 32)
	output:	(None, 12, 12, 32)
conv2d_3: Conv2D	input:	(None, 12, 12, 32)
	output:	(None, 12, 12, 32)
up_sampling2d_1: UpSampling2D	input:	(None, 12, 12, 32)
	output:	(None, 24, 24, 32)
dropout_2: Dropout	input:	(None, 24, 24, 32)
	output:	(None, 24, 24, 32)
conv2d_4: Conv2D	input:	(None, 24, 24, 32)
	output:	(None, 24, 24, 32)
up_sampling2d_2: UpSampling2D	input:	(None, 24, 24, 32)
	output:	(None, 48, 48, 32)
conv2d_5: Conv2D	input:	(None, 48, 48, 32)
	output:	(None, 48, 48, 1)

Figure 2. AutoEncoder architecture used

2.2. UNET

The AutoEncoder model can preserve the dimensionality of the image, but the linear comparison of the input leads to a bottleneck that doesn’t transmit all features. However, the UNET overcomes this limitation by adding a skip connection that allows feature representations to pass through the bottleneck. UNET was first introduced and used for Biomedical Image Segmentation [2], but it can also be used for image denoising and other image processing tasks. The architecture of the UNET model used for the experiment is shown in Fig. 3.

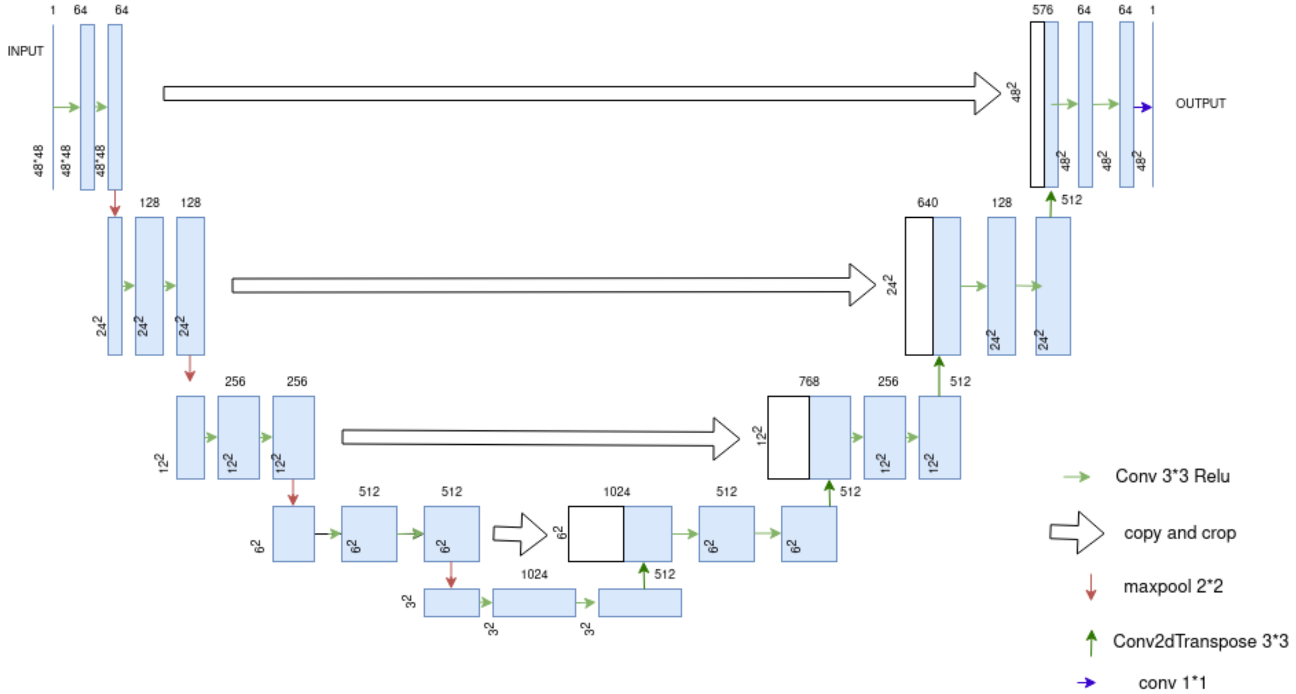


Figure 3. Architecture UNET (modified)

Certain changes in the original architecture [2] are done as per the requirement while experimenting.

3. Materials and methods

FER2013 dataset [5] is used for training and testing purposes. It consists of 35,000 images of different facial expressions. The dataset is split into a 80:20 ratio; 80% of the total data for training and remaining for testing purposes. About 100 images are kept aside for the final testing purposes.

3.1. Programming

Our study involves Keras’sDeep Learning framework which is used to build the model. Different types of noise are introduced in the image to produce a noised image. The noise and noise-free images are used to train and test the model.

3.2. Metrics

Two metrics are mostly used to evaluate image denoising tasks, PSNR and SSIM. PSNR stands for Peak signal-to-noise ratio. The equation of the PSNR is shown in (1).

$$PSNR = 20 - log_{10}(MAX_I) - 10 * log_{10}(MSE) \tag{1}$$

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

MSE stands for mean square error. Its mathematical representation is shown in equation(2). The m*n represents noise-free monochrome image I having ‘K’ as noise approximation. MAX_I is the maximum pixel values per pixel.

SSIM stands for structural Similarity. The PSNR is not highly indicative of the perceived similarity of the image. So, SSIM is used to address the shortcoming by taking texture into account. Equation (3) is the mathematical representation of SSIM.

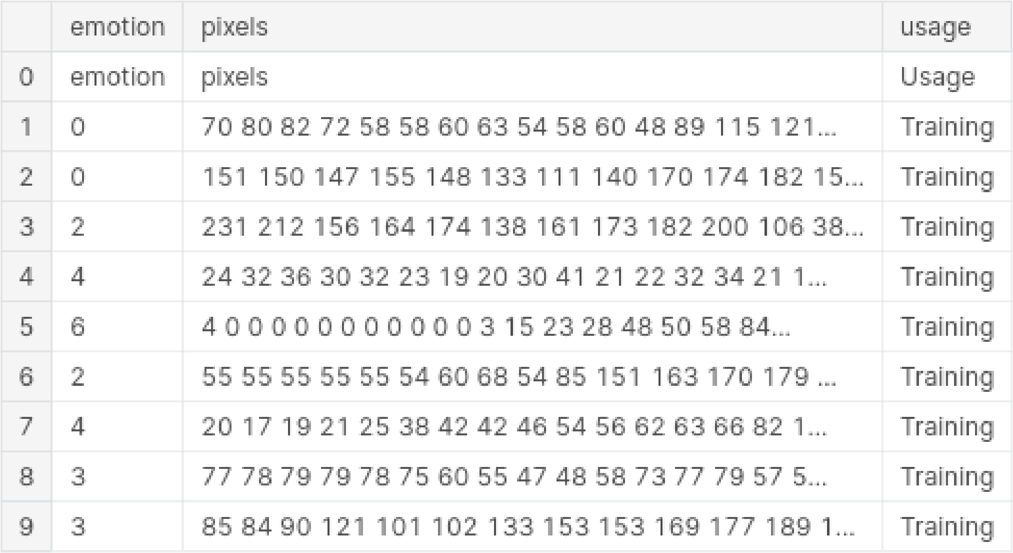
$$Q = \frac{4\sigma_{xy}\bar{x}\,\bar{y}}{(\sigma_x^2+\sigma_y^2)[(\bar{x})^2+(\bar{y})^2]} \tag{3}$$

$$Q = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2+(\bar{y})^2} \cdot \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2} \tag{4}$$

4. Results

4.1. Generation of noisy images

Initially, images are gained as pixel value as they are represented in this form. Fig. 4 shows the sample of the dataset. Taking those pixel value images are generated and resize into 48*48 pixels. The images are split into 80:20 for training and testing purposes. Each image is normalized by dividing each pixel value by 255.0 For each training and testing data, the same process is taken, and add different types of noise are added later to generate the noised dataset. Gaussian, Salt&Pepper, and Poisson Noise are added to the image with a certain random noise value. Noised images are generated by overlapping the image with a certain noise factor from range (0.01, 0.1) and then clipped between (0,1). To generate noised images Skimage and NumPy random normal distribution is used. The process for the addition of Noise is shown in Fig. 5. The image gained after preprocessing can be seen in Fig. 6.



	emotion	pixels	usage
0	emotion	pixels	Usage
1	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
2	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
3	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
4	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
5	6	4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training
6	2	55 55 55 55 55 54 60 68 54 85 151 163 170 179 ...	Training
7	4	20 17 19 21 25 38 42 42 46 54 56 62 63 66 82 1...	Training
8	3	77 78 79 79 78 75 60 55 47 48 58 73 77 79 57 5...	Training
9	3	85 84 90 121 101 102 133 153 153 169 177 189 1...	Training

Figure 4. Dataset sample

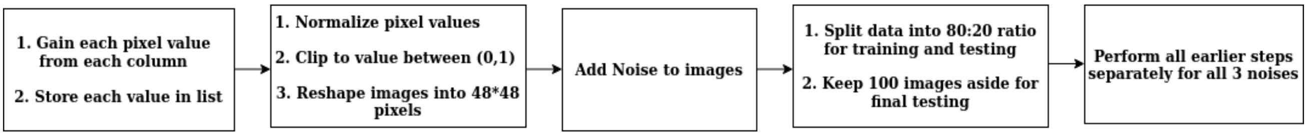

1. Gain each pixel value from each column
2. Store each value in list
1. Normalize pixel values
2. Clip to value between (0,1)
3. Reshape images into 48*48 pixels
Add Noise to images
1. Split data into 80:20 ratio for training and testing
2. Keep 100 images aside for final testing
Perform all earlier steps separately for all 3 noises

Figure 5. Image normalization, Gaussian noise addition, and clipping

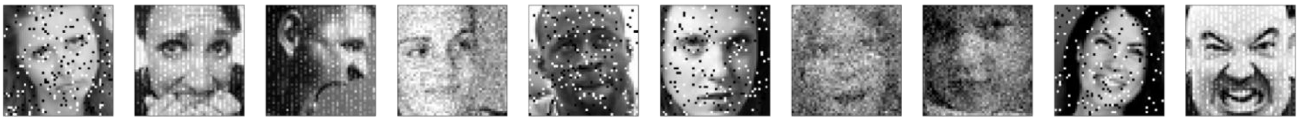


Figure 6. Dataset visualization

Equation (5) represents the probability density function of the Gaussian random variable z. The grey level is represented by ‘z’, grey value is represented by ‘μ’ and ‘σ’ represents standard deviation.

$$\rho_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{5}$$

Equation (6) represents the Poisson distribution. The ‘x’ represents the whole number, ‘γ’ represents the mean of several occurrences in the interval and ‘e’ is the Euler’s constant.

$$P(X = x) = \frac{\gamma^x e^{-\gamma}}{x!} \tag{6}$$

Equation (7) represents the salt and pepper distribution function. If P_a & P_b= 0 then it is a bipolar impulse or salt and pepper noise.

$$p(z) = \begin{cases} P_a \ for\, z = a \\ P_b \ for\, z = b \\ 0 \ \ otherwise \end{cases} \tag{7}$$

4.2. Training and evaluation

Secondly, both models are trained in training data to generate a clear noise-free image. Training is carried out on 35 epochs with mini-batch batches of 64 images per batch, but the early stopping criteria usually ends training earlier. Adam [11] optimizer used and MSE is used as loss criterion. Table-1 represents hyperparameters being used in more detail. One thing that can be noticed easily is that the epochs for AutoEncoder and UNET is different. This is gained after experimentation, but the early stopping condition stops training earlier than provided epochs. The hyperparameters used while training both models can be seen in Table 1.

Table 1. Model hyperparameters

Parameter	Values
Optimizer	Adam
Epochs(Autoencoder)	35
Epochs(UNET)	10
Batch Size	64
Learning Rate	0.001
Loss	MSE

Thirdly, the model is used to generate a clear image using noised test image, and PSNR and SSIM between the original image and generated image are calculated. Both gained values are better in the UNET model than the basic AutoEncoder model. While increasing the basic model size the model is unable to generate a better image, so the UNET model is used which can generate better images with optimal PSNR and SSIM values. The comparison of PSNR and SSIM values between AutoEncoder and UNET is shown in Table 2 and Table 3 respectively. In all three cases, the UNET based model has performed better than AutoEncoder by a large margin.

Table 2. Comparison PSNR value dealing with noise factor of 0.05

	Gaussian Noise	Poisson	Salt and Pepper
Basic Encoder	26.589794	27.308831	26.004077
UNET	31.343941	35.099072	36.791870

Table 3. Comparison SSIM value dealing with noise factor of 0.05

	Gaussian Noise	Poisson	Salt and Pepper
Basic Encoder	0.976593	0.980438	0.973197
UNET	0.998250	0.997029	0.998480

4.3. Visualizing output

Finally, both the basic AutoEncoder and UNET reduce the noise of the noised image. Both the gained generated image is evaluated by using the performance metrics. The UNET model can decrease the noise more compared to the basic AutoEncoder. Using the MSE as criteria both models can reduce the noise. Fig. 7-9 shows the visualized output of the model.

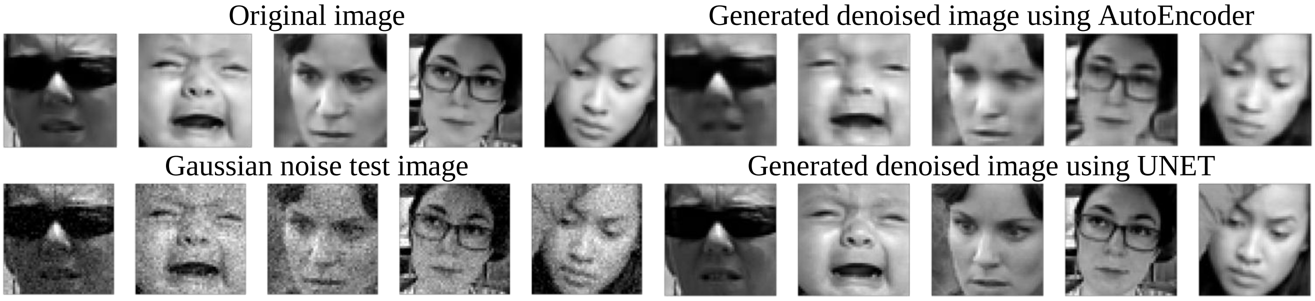

Original image
Generated denoised image using AutoEncoder
Gaussian noise test image
Generated denoised image using UNET

Figure 7. FER2013 images (a) original image (b) Gaussian noise image with noise factor of 0.05 (c) denoised image with AutoEncoder (d) denoised image with UNET

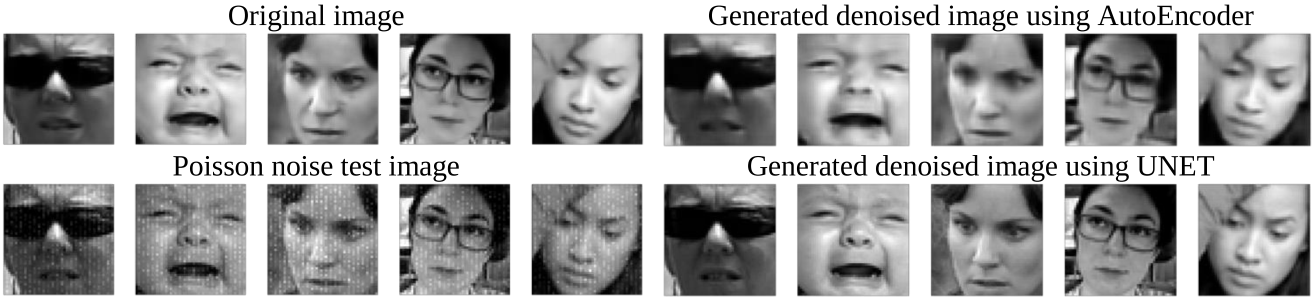

Original image
Generated denoised image using AutoEncoder
Poisson noise test image
Generated denoised image using UNET

Figure 8. FER2013 images (a) original image (b) poisson noise image with noise factor of 0.05 (c) denoised image with AutoEncoder (d) denoised image with UNET

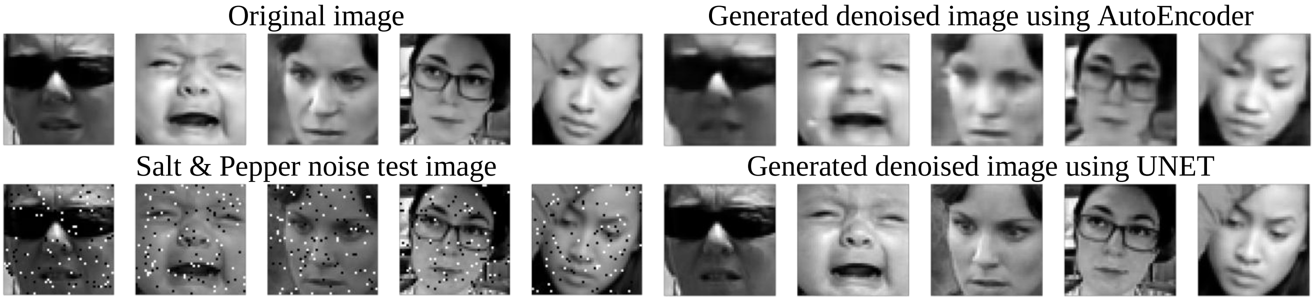

Original image
Generated denoised image using AutoEncoder
Salt & Pepper noise test image
Generated denoised image using UNET

Figure 9. FER2013 images (a) original Image (b) Salt&Pepper noised image with noise factor of 0.05 (c) denoised image with AutoEncoder (d) denoised image with UNET

5. Conclusion

Our experiment results show that the UNET model can denoise images better than the AutoEncoder. The gained PSNR and SSIM clearly show that UNET is far better than the basic AutoEncoder. The gained denoised image from the AutoEncoder is a blurry one which is seen in the generated images and the hypothesis that the skip connection in UNET will resolve the issue is also visible in the result. The results clearly show compared to basic AutoEncoder and even larger ones, UNET is a far more effective and efficient way to denoise images. Other than denoising, the UNET model can also be used for image deblurring, image restoration, etc.

Further, an automatic system can be built which automates the task of classification of the noise when a noised image is passed to the model. This addition will greatly increase the automation in process of denoising.

References

[1] L. Fan, F. Zhang, H. Fan, and C. Zhang, “Brief review of image denoising techniques,” Visual Computing for Industry, Biomedicine, and Art, vol. 2, no. 1. 2019, doi: 10.1186/s42492-019-0016-7.

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[3] I. Ali et al., “Image Denoising with Color Scheme by Using Autoencoders,” Ijcsns, vol. 18, no. 12, pp. 158–161, 2018.

[4] Latha H N and Rajiv R Sahay, “A Local Modified U-net Architecture for Image Denoising,” Int. J. Mod. Trends Sci. Technol., vol. 6, no. 8S, pp. 140–144, Sep. 2020, doi: 10.46501/IJMTSTCIET27.

[5] “fer2013 | Kaggle.” https://www.kaggle.com/deadskull7/fer2013 (accessed Jul. 12, 2021).

[6] A. Awad, “Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise,” Eng. Sci. Technol. an Int. J., vol. 22, no. 3, pp. 746–753, Jun. 2019, doi: 10.1016/J.JESTCH.2019.01.012.

[7] S. Ghose, N. Singh, and P. Singh, “Image denoising using deep learning: Convolutional neural network,” Proc. Conflu. 2020 - 10th Int. Conf. Cloud Comput. Data Sci. Eng., pp. 511–517, Jan. 2020, doi: 10.1109/CONFLUENCE47617.2020.9057895.

[8] O. Sheremet, K. Sheremet, O. Sadovoi, and Y. Sokhina, “Convolutional Neural Networks for Image Denoising in Infocommunication Systems,” 2018 Int. Sci. Conf. Probl. Infocommunications Sci. Technol. PIC S T 2018 - Proc., pp. 429–432, Jan. 2019, doi: 10.1109/INFOCOMMST.2018.8632109.

[9] J. Gurrola-Ramos, O. Dalmau, and T. E. Alarcon, “A Residual Dense U-net Neural Network for Image Denoising,” IEEE Access, 2021, doi: 10.1109/ACCESS.2021.3061062.

[10] C. Tian, Y. Xu, and W. Zuo, “Image denoising using deep CNN with batch renormalization,” Neural Networks, vol. 121, pp. 461–473, Jan. 2020, doi: 10.1016/J.NEUNET.2019.08.022.

[11] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” undefined, 2015.

[12] Z.Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[13] J. Korhonen and J. You, “Peak signal-to-noise ratio revisited: Is simple beautiful?,” 2012 4th Int. Work. Qual. Multimed. Exp. QoMEX 2012, pp. 37–38, 2012, doi: 10.1109/QOMEX.2012.6263880.

[14] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Networks, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/J.NEUNET.2014.09.003.