SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Autoencoder-based image denoiser suitable for image of numbers with high potential for hardware implementation

Mahmoud Maghraby[1], Nahla Elazab Elashkar[2,*]

[1] Electrical and Communication Engineering Department, Faculty of Engineering, British University in Egypt, Egypt
[2] Microelectronics Department, Electronics Research Institute (ERI), Cairo, Egypt

ARTICLE INFO

Article history:
Received 10 October 2023
Received in revised form 20 December 2023
Accepted 10 February 2024
Available online 5 May 2024

ABSTRACT

The algorithms for processing images and videos are currently essential for many applications. Many of these applications are specified for processing and analyzing images of numbers, such as smart meter reading, automated document processing, and processing of vehicles and license plate images in traffic monitoring and analysis. Consequently, eliminating noise is frequently used as a pre-processing step to improve subsequent analysis and processing outcomes. In this context, this manuscript proposes using artificial intelligence-based methods to increase the efficiency of the image-denoising process. However, the computational demands of these algorithms necessitate careful consideration of the hardware on which they are implemented. Therefore, this paper proposes using the simple autoencoder approach and evaluates its efficiency compared to the conventional methods. This unsupervised model is trained to identify and remove impulse noise from digital images by replacing some pixels with others from the outer dataset that can clarify the whole image more. The model was trained using handwritten numbers, MNIST, and data set size in the first trial and the FER2013 dataset in the second. The model is superior in the case of the simple dataset. Two versions of autoencoders are considered, the first with three layers and the second with five. The Traditional denoising methods are investigated for comparison purposes. The four conventional filtering procedures, AMF, DBMF, ADBMF, and MDBUTMF, are implemented using the MATLAB simulation environment, and the results are reported and compared with the proposed methodology. The results show that the proposed artificial intelligence-based method significantly outperforms the traditional methods regarding processing efficiency and the resulting image quality. Moreover, the computational intensity for the proposed methodology is chosen as a metric for evaluating the algorithm compliance for the hardware implementation compared to the other Artificial Intelligence (AI)-based denoising algorithms. The suggested technique has minor processing and training time compared to the other AI-based methods with adequate quality in case the images of numbers usually do not contain many details, making it more convenient for hardware implementation.

Keywords:
Image Denoising; Convolutional Neural Network; Digital Image Filtering; Salt and Pepper Noise

* *Nahla Elazab Elashker.*
E-mail address: nahlaelazab@eri.sci.eg

https://doi.org/10.37934/araset.44.2.234246

1. Introduction

Digital image processing, in general, plays a significant role in our lives nowadays. It is used in many applications, such as face detection and recognition, image reconstruction, traffic sensing technology, and some medical applications. The images of numbers have numerous applications, such as monitoring and analysis of vehicles and license plates, real-time data analytics for images of physical objects that contain numerical information, and automated document Processing for images containing numbers, such as invoices, receipts, or forms. These images go through several phases before they are used by the user, including image enhancement, compression, restoration, etc. Image enhancement, in which the image is denoised, is one of the most crucial steps since the images commonly contain a wide variety of noise. For example, digital photos exhibit Gaussian noise due to electrical and sensor noise due to poor lighting and high temperatures. Generally, three main forms of noise are frequently produced by cameras: random noise, banded noise, and impulse noise, which is known as salt and pepper noise [1]. Due to these noises, the digital image must be filtered. There are several traditional filter types: Bilateral filter, Median filter, Mean filter, and Gaussian filter, etc. Some filters are used to decrease the impulse noise, starting with the Standard Median Filter (SMF), the basic algorithm to remove the noise from the image to more advanced ways and algorithms. Different filtering techniques are better suited for different types of noise than others. For example, the median filter is best suited for salt and pepper noise, while the mean filter is best suited for Poisson noise. As a result, the Gaussian noise is eliminated using the Gaussian filter. These filters are mainly classified into two groups: spatial and frequency domains. Spatial filtering is much simpler than frequency filtering. In the spatial domain, there are two primary types of noise reduction methods: linear filtering and nonlinear filtering techniques. Because of the low pass feature, linear filtering algorithms smooth down the edges in images and remove abrupt, sharp transitions in the initial data. The image may become blurry as a consequence. As a result, the retrieved result could not be adequate. So, usually, nonlinear algorithms are used more frequently for picture filtering. The median filter is one of the most popular conventional filters and falls under nonlinear spatial techniques. That is because impulse noise generates abrupt and harsh perturbations in the visual signal [2], [3]. These traditional approaches provide accurate results as long as noise density is low, but techniques should be modified to restore the image in the case of high noise density. First, more focus was placed on hybrid filters resulting from a mixture of the median and mean filtering concepts. Further along, the rise of Artificial Intelligence (AI) and machine learning have introduced new methods and techniques to accomplish the image-denoising process with more efficient results. Some of their main ideas come from the fact that most noise corrupting the image has statistical patterns and properties that the network can detect and determine [4]. There are 136 billion photos on Google; this vast number can be used as reference data to improve image filtering methods. Initially, an AI algorithm for denoising and resolution augmentation is developed by Subtle Medical Company to enhance the existing MRI scanners [5]. As the leading cause of image corruption is during image acquisition, and this depends on the quality of the image sensors, the low-noise images need professional cameras, which are usually expensive. However, recently, the noise from ordinary cameras can be removed using AI, object detection, and computer vision algorithms. Even in the medical section, improving the digital image processing denoising techniques, specifically AI-based ones, can reduce the cost of medical scan instruments, e.g., X-ray and Magnetic Resonance Imaging (MRI)[6]. However, using unnecessary intensive computational techniques for highly needed and widespread applications, such as those dealing with numbers and simple images, may be discourageable for hardware implementation. In this case, the only required features beyond the high accuracy are the fast processing time and the computational simplicity that guarantees the possibility to be hardware implemented [7], [8]. In light of the discussion mentioned above, this research aims to propose a simple autoencoder-based image denoiser with three and five layers and also discuss and investigate the conventional image filtering techniques to be a reference for evaluating the efficiency of the proposed technique. The paper will focus on the various versions of the median filter as it is one of the commonly used filters because of its ability to remove impulsive noise. The suggested AI-based model is utilized to reach a dataset of the targeted image and then modify the corrupted image using different images of the same dataset, aiming to improve the efficiency of digital image processing in the presence of this noise. Additionally, more complicated AI-based techniques such as CNN with median layer, noise2noise technique, and generative adversarial network technique are studied [9]. Their performances are evaluated and compared to the suggested technique to assess the computational density in terms of the processing time, which gives an insight into the hardware implementation's applicability. This paper is organized as follows: Section 2 demonstrates the traditional methods of impulse noise filtering and their performance measurements, which will be considered as a reference to rate the performance of the proposed technique. The AI-based denoising techniques are reported in Section 3. The details of the presented autoencoder-based denoiser are depicted in Section 4. Finally, the paper is concluded in section 5.

2. Denoising using traditional Impulse Noise Filters

Digital image Noise is always defined as unwanted information affecting the desired image. In the case of the impulse noise (salt and pepper), the noise takes only presence as a pixel having a maximum or minimum value, and those values are 0 or 255. One of the most common conventional ways to reduce salt and pepper noise from the corrupted image is the Standard Median Filter (SMF), which replaces the distorted pixel by another with a closer value. This technique usually forms a 3x3 window or mask where the targeted corrupted pixel is in the middle, and the values of the 8-neighboring surrounding pixels are rearranged. The median of their values is considered the near value for this corrupted pixel. Due to the low efficiency of the SMF at high noise densities, there are modified versions of it, such as Adaptive Median Filter (AMF), which is beneficial for filtering an image with high noise density. Unlike the standard median filter, the adaptive median filter changes its sub-image window size according to the noise in the window. The image sub-window could start an initial value of 3x3, and this can be considered the first cycle of filtering. The sub-image window size is increased to 5x5 and could be regarded as the second filter cycle if noise is still found. Another example is the Decision-Based Median Filter (DBMF), Its algorithm is slightly similar to the SMF, but instead of letting the sub-image window go through the whole image, it only works when detecting a corrupted pixel and putting it at the center of the sub-image window and replace it by the median value of the sub-image window [10], [6]. At the same time, it preserves the noise-free pixel value. A combination between the AMF and DBMF is the Adaptive Decision-Based Median Filter (ADBMF). This algorithm has higher efficiency than the SMF as it leaves the uncorrupted pixels untouched and the sub-image window value changes according to the number of corrupted pixels in the window [11].

Regarding high noise density, SMF and DBMF efficiency decreases to very low levels. So, one of the solutions is the Modified Decision-Based Unsymmetric Trimmed Median Filter (MDBUTMF) which operates with high noise densities of 80% to 90% [12]. By importing the noisy image and then checking each pixel, the method is made to remove the excessive noise density [13]. Like the decision-based median filter, nothing changes if the pixel is not damaged. However, if the pixel was damaged, check the window. The pixel is removed and replaced with the window's mean value if the window had all corrupted pixels. However, if the window had any uncorrupted pixels, the damaged pixels would be eliminated, and the pixel would then be replaced with the median value of the window's remaining pixels. The image is restored using this technique in a way that is quite effective at getting the image as near to the original as feasible [14]. The four filtering procedures (AMF, DBMF, ADBMF, and MDBUTMF) are implemented using the MATLAB simulation environment by storing a clean 256x256 and then adding salt and pepper noise with different density ratios then using the algorithm under test for removing this noise. The result of applying these techniques on a sampled image with 40% noise density is presented in Figure. 1. Furthermore, the filtered and original images are compared to evaluate the technique's performance using different criteria, as shown in Figure 2 and Figure 3. The Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) are used to assess filtering procedures as illustrated in Figure 2 and 3, respectively. PSNR calculates the ratio of the maximum possible power of the signal (in this case, the maximum pixel value) to the power of the difference between the filtered and ground truth images. It is often expressed in decibels (dB) and measures image fidelity. It is the main metric that will be used for all the investigated and proposed techniques in this work. Also, the Mean Squared Error (MSE) is calculated as a second metric and to get the PSNR value, giving the PSNR evaluation an edge over other evaluation metrics by including the MSE value.

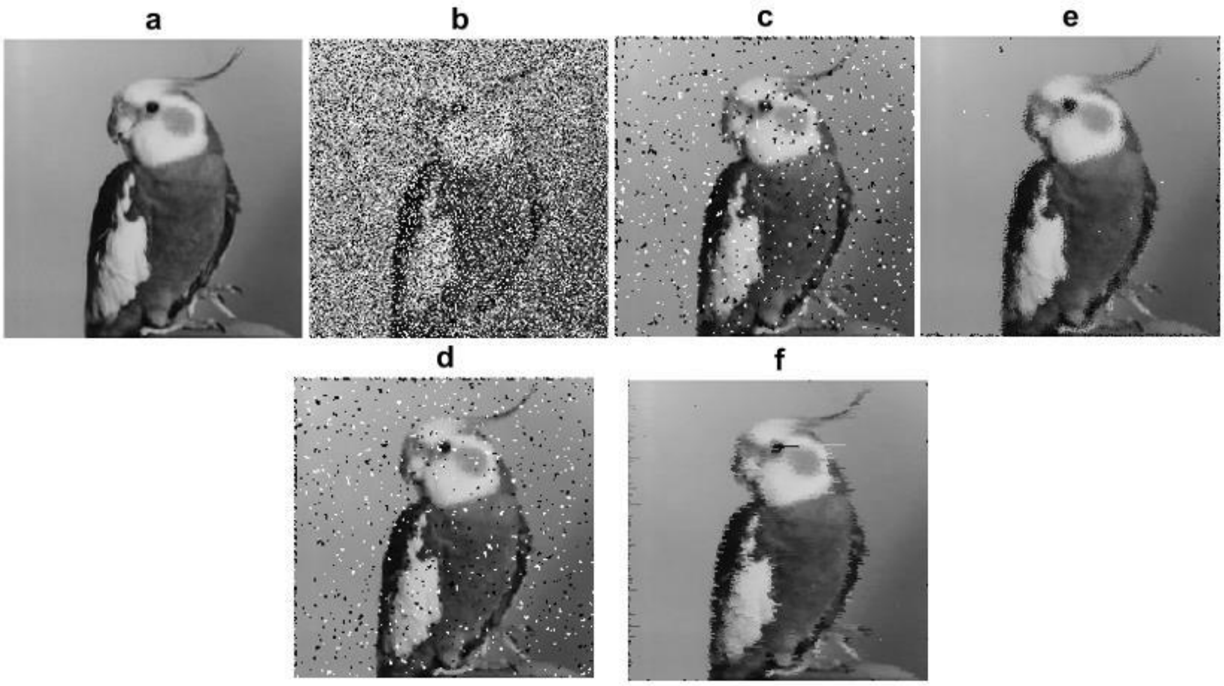

Fig. 1. (a) original image (b) original image with 40% noise density (c) DBMF result (d) SMF result (e) MDBUTMF result (f) ADBMF result.

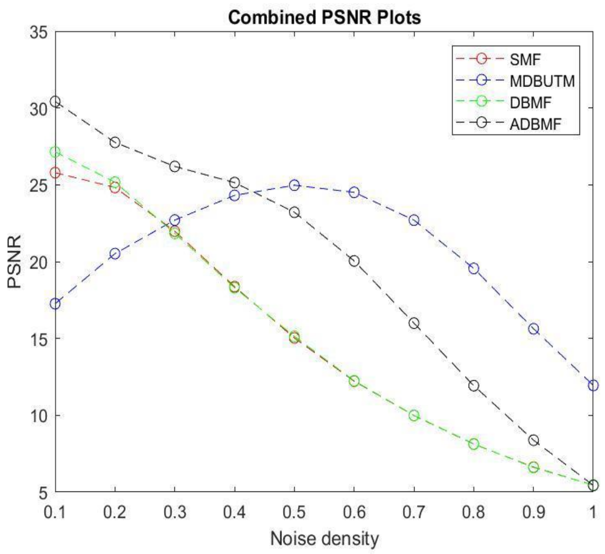

Fig. 2. PSNR versus noise density for AMF, DBMF, ADBMF, and MDBUTMF algorithms

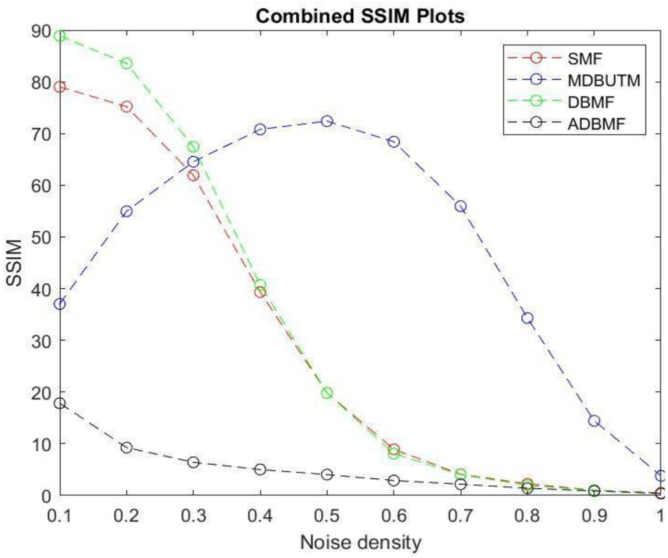

Fig. 3. SSIM versus noise density for the AMF, DBMF, ADBMF, and MDBUTMF algorithms

The higher PSNR value the higher the efficiency of this filtering technique. The PSNR for a grayscale image is calculated using Eq. (1)

$$PSNR = 10 \log (255^2/MSE), \quad (1)$$

where MSE is the Mean Squared Error that can be calculated using Eq. (2)

$$\mathrm{MSE} = \Sigma(o(i,j) - f(i,j))^2/(m \times n), \quad (2)$$

where m and n are the size of the image row and column. While (o) is the original image and (f) is the filtered one. In an RGB image, the average PSNR can be calculated as the sum of the PSNR of each color plane divided by three. The SMF is considered the basic technique that has low efficiency in the case of high noise-density images. Another problem emits as the sub-image window filters the whole picture; in the case of using SMF with a low noise density image, the image loses its sharpness. While using AMF or DBMF could result in slighter sharpness loss and higher accuracy. At the same time, ADBMF, a combination of adaptive and decision-based filters, removes the salt and pepper in the image but it is still ineffective against high noise density. The relation between PSNR and the noise density while using the four different filtering methods indicates that no perfect filter is suitable for all noise densities. But the superiority depends on the purpose of the filtering process itself. In high noise density, it is better to choose the MDBUTMF as the ADBMF leaves the value of the noise-free pixel without changing. Briefly, this section investigates the conventional filtering techniques and finds out that the brightness and structure of the image are impacted by the ADBMF But they are unaffected by the DBMF and SMF, in contrast. The adaptive decision-based median filter provides a significantly higher PSNR than SMF and DBMF when comparing their PSNR values. Additionally, the ADBMF has the highest PSNR value at 40% noise density when compared to the MDBMF. The MDBMF, however, has the highest value when SSIM is measured, with a noise density of 40%. Afterwards, the next sections demonstrate the utilization of the AI in noise filtering purpose.

3. Denoising using Artificial Intelligence

The ability of machines to carry out activities that ordinarily require human intelligence, such as comprehending natural language, recognizing objects, making judgments, and learning from experience, is known as artificial intelligence (AI). AI technologies rely on algorithms and mathematical models to analyse large amounts of data [15].

This section will discuss the processing of some neural network models to the image data and demonstrates the image denoising using them. Some of these models are tested using the used dataset

3.1 The concept of convolutional neural network (CNN)–based denoiser

CNN has a scheme for detecting and analyzing patterns. The simplest CNN consists of three main layers; the input, output, and hidden layer which connects the input and the output layers. It receives input, transforms it using the hidden layers, and then repeatedly sends it to the next layer hidden layer, if more than one, until reaching the final (output) layer. In each layer, the filter for this layer should be specified. According to the filter, the transformed input of this layer and this filter is responsible for detecting the patterns, shapes, or edges of the image. AS the more hidden layers are added, the more features the network can detect. The filter of a CNN layer is usually applied to the image using a 3x3 window passing through the whole image, as shown in Figure 4. In this research work, the ability of the model to detect the object is used for image denoising. A handwritten image dataset is used. For detecting which number is written; first, the data is imported to the proposed network and split into train and test groups, where the train image group represents the data that is given to the network to learn from. The test group is the remaining data to check the model's accuracy. Three different networks are used, each network has its own filter, and the output of this filter is a detected feature. The result of those features can conclude the desired output. The difference in each layer can be observed due to the different filters applied on this layer. Figure.4 displays the inputs and outputs of CNN using different filters.

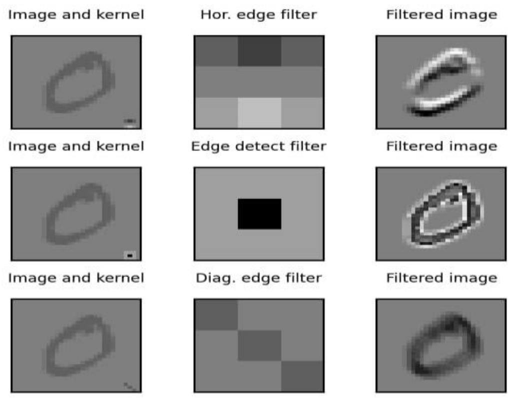

Fig. 4. the results of passing the image through the CNN network

Recently, for training the model to denoise the image, the neural network should be trained on two datasets, one for the clear images and the other containing the same images with same arrangement but corrupted by noise. So the model can find a relation and pattern between the plain image and its corresponding noisy image. In contrast, the most traditional AI based method is training the neural network using pairs of data containing clean and noisy versions of the same image to detect the noise and replace the corrupted pixels with the correct ones from the clean set. Therefore, many approaches for removing the noise from images using neural networks like convolutional neural networks with median layer, noise2noise image denoising, generative adversarial networks, and autoencoders [16]. These approaches are still being developed to denoise the image and increase the image quality. The first three models are employed, and their outputs are assessed and contrasted with a recommendation of the last one for hardware implementation in case of simple processing input data as explained, in detail, through the following subsections.

3.1.1 CNN with median layer for denoising image

The concept of the CNN median layer is to combine the standard CNN layers with median filter layers, so the design of this network is fully CNN injected with median filter layers between the hidden layers that cause a difference instead of using the median filter directly on the corrupted image. So as the network layers are extracting features from the image, the median layer is being applied in some of these feature channels that causes removing noise in different features channels output and add the denoising features to predict a higher quality image. That is because the median layer acts in this process as the usual way of removing noise, allowing the noise-free pixels to pass clearly. This median layer can be defined as the median filter applied on each element of the feature channels as the basic concept of mask scanning through the whole matrix. This median layer is applied to every layer in the neural network, so if we have ten layers, there is an additional median layer for each. This method is considered to be fully CNN. The network does not have any limitations on the size of the input. It begins with two back-to-back median layers, followed by residual blocks and median layers in a specific order. At the end of the network, there are residual blocks without any median layers inserted between them. In the implementation stage of this network, the median filter layer is applied on the first half of the layer, so the first half of the network is designed to remove the noise from the image. In contrast, the second half is designed for recovering the input image. This method was designed to generate 64 features per layer.

The residual block is used to skip the connection over convolutions and includes normalizing batch layers. It is a type of building blocks used in deep neural networks, specifically in CNNs. They were first introduced in the ResNet (Residual Network) architecture to help solve the problem of vanishing gradients in deep networks [17]. A residual block comprises several layers of convolution, batch normalization, activation functions, and a shortcut connection that skips one or more of these layers. The idea is that the network learns a residual function representing the difference between the original layers' output and the shortcut connection input. This residual function can then be added back to the shortcut connection output, which helps bypass the problematic layers and allows the gradient to flow more easily. The loss in the image is defined as the MSE of the estimated image and the original image. MSE and PSNR are inversely proportional; a lower MSE means higher image quality after denoising. Figure 5 illustrates the CNN median layer, which consists of 64 layers; between each layer, a median layer acts as a median filter [16].

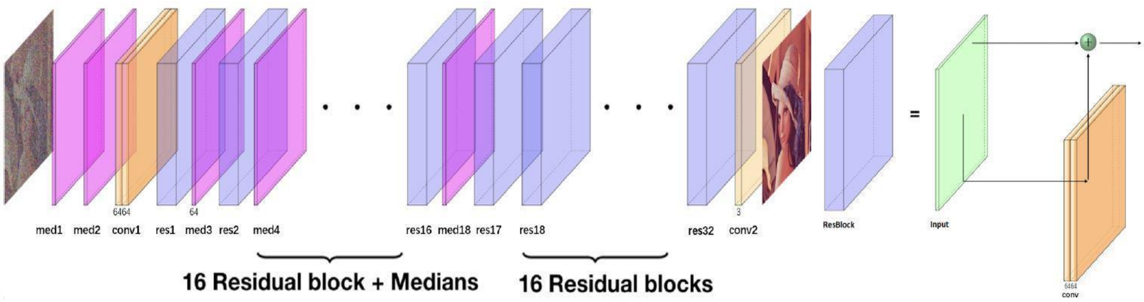

Fig. 5. CNN median layer [16]

To test this network, a dataset of 90 200x200-pixel photos was used to train the model. The network's output is a 70x70-pixel scaled image that contains the image's denoised version. The model extracts a set of weights in layers that can convert an input noisy image to a clean image during training. Three separate sets of photographs were compared in order to gauge and assess this technique's performance in contrast to other techniques. The well-known digital image processing is included in the first set. The Kodak Image Dataset is the third collection, while the second set is BSD300. Both sets were denoised using CNNs. PSNR is the statistic taken into account in the comparison. First, train three pairs of fully convolutional networks, one with median layers and the other without them, with most networks composed of residual or convolution blocks. Next, train two sets of deep fully convolutional networks, one of which is a conventional network devoid of any median layers and the other is a network that is the exact opposite of the first set but with median layers; this will help to illustrate the benefit of the median layer. Median layers are present in the first half of the network but not in the second. The networks in the first set consist of repeating residual, batch normalization, and activation, or convolution blocks. The results demonstrate how median layers boost the network's PSNR value while causing fewer training-related losses than the other two methods. Additionally, the findings demonstrate that networks with the median layer typically have greater PSNR values than other networks with values between 0 and 5dB. The calculating PSNR for the Median layer CNN at different noise densities is tabulated in Table 1 showing very good performance, using RGB images of size 256x256 for the salt and pepper noise. However, this architecture is considered a complex model that is suitable for denoising of the images of significant details not for the simplest ones as the images of numbers and letters. Hence, it is not recommended for the hardware implementation of the image of numbers processing applications.

Table 1
The calculating PSNR for the Median layer CNN at different noise densities

Noise density %	Median layer CNN
30%	37.04 dB
50%	35.00 dB
70%	33.07 dB

3.1.2 Noise2Noise based denoiser

The noise2noise technique applies basic statistical reasoning for image enhancement and signal reconstruction by using machine learning to map corrupted pixels to clean pixels by using a method where the model can restore images by only looking at corrupted images. In the traditional image denoising techniques, we can observe the usage of statistics to replace the noisy pixel with a suitable pixel value. Recently, with the progress of AI and deep learning, it showed a potential to replace the old ways. Some of AI methods require introducing a clean and noisy version of the same image so the model can map the differences. While in the noise2noise technique, a pair of the same image is used but with different noisy intensities and starts extracting the features from those images. This approach is called self-supervised learning, where the model receives unlabeled data and starts processing it without human intervention. So the model can learn how to remove the noise when receiving a new noisy image with the same features. The noise2noise trains the model to learn the noise pattern itself, as most noise has statistical properties. The noise2noise model tries to detect these statistical patterns in the noisy image and remove them. This method is considered one of the latest methodologies to remove noise. However, its flaw appears when receiving a noisy image with a different statistical pattern; its learning will not be effective. Since the noise2noise technique focuses on the type of noise pattern, the dataset must contain many noise patterns to enable the model to detect different noisy images, making the model more immune to different noise patterns [18]. The most challenging thing in this technique is the loss function and how to decrease it as much as possible. Although the Noise2Noise has many promising upgrades, it has some drawbacks that make other techniques still ahead of it. It can fail in overfitting the noise pattern if the input image has a different one; at his time, the model's accuracy will drop badly. Also, the Noise2Noise requires very high processing, especially when the training dataset is extensive, or in the case of RGB images rather grayscale. It might be an arduous processing technique which makes it inconvenient to the hardware implementation. The calculating PSNR for the Noise to Noise-based CNN at different noise densities is tabulated in Table 2 using RGB images of size 256x256 for the salt and pepper noise.

Table 2
The calculating PSNR for the Noise to Noise-based CNN at different noise densities

Noise density %	Noise2Noise
30%	36.39 dB
50%	34.68 dB
70%	32.83 dB

3.1.3 Generative Adversarial Network

Generative Adversarial Network (GAN) was introduced by Ian Goodfellow in 2014 [19]. The GAN first concept was to train the model on specific data and to make it able to generate similar data. For example, suppose the model was given a dataset containing a particular artist's paintings. In that case, the model will try to generate an image containing a painting that looks close to this artist's drawings. As mentioned, the GAN was used firstly to generate new data with a first model called the discriminator and a second model called the generator. The discriminator is trained on particular data to detect whether any other data belongs to it. As in the painting example, the discriminator is trained on Van Gogh's painting to detect if any other painting belongs to him or not, while the generator tries to assign values to pixels and then send them to the discriminator to decide if this image belongs to Van Gogh paintings or not. Suppose the discriminator decides it does not belong to Van Gogh's painting. In that case, the generator tries again until the discriminator loses its ability to detect any difference and is tricked by the generator. At this point, the generator can copy the drawing style of Van Gogh. In denoising images using GAN, the generator network receives a noisy image while the discriminator has a clean version of the image. Then the generator network tries to denoise the image until the discriminator accepts it. This whole process is repeated on a dataset containing clean images and another noisy version of these images. An example of using GAN for denoising the image is GAN2GAN, where the primary motivation of GAN2GAN is simple; given a single noisy image Z(i), we want to generate two image pairs ($\hat{Z}_1$ (i), $\hat{Z}_2$ (i)) that correspond to the noisy versions for the same underlying clean image of Z(i), but each with independent realization of the noise. Such generation is challenging since there is a necessity to blindly separate the noise, the clean image solely from Z(i), learn the distribution of the noise, and switch only the noise part of Z(i) with the independent realizations of the noise [20]. Despite the challenge, those pairs of images can be used to carry out the N2N training to train a denoiser as soon as it is successful.

Furthermore, the noisy dataset should contain different types of noise and different noise levels so the model can be trained to denoise different types of noises with different densities. The GAN denoising method has many advantages and disadvantages, since the output generated by applying a trial and error concept until the discriminator accepts the results. So, the denoised image output will also have some realistic features. This technique is more flexible to denoise different types of noisy images. On the other hand, training the GAN model is much more complicated than training a traditional denoising CNN. Also, the generator model can collapse and may not give an acceptable result due to the inappropriate dataset. So the dataset must be chosen very wisely. The GAN methodology requires much more processing time than the regular denoising convolution networks because of the network architecture. As a result, it is also not recommended for hardware implementation of denoiser for number images.

4. The proposed Autoencoder-based denoiser

The autoencoder is one of the most accessible techniques to understand, as it has a straightforward network architecture. It is considered an unsupervised neural network whose primary purpose is to copy input to output. The autoencoder consists of three main layers: the encoder, decoder, and latent space are illustrated in Figure. 6. The encoder layer receives the input dataset, then reduces its dimensions and compresses it by downsampling this input data. The decoder layer receives the compressed data and then reconstructs it to become as close as the original the decoder layer so it can be considered the opposite of the encoder layer. At the same time, the latent space or the bottleneck layer is the connecting chain between the encoder and the decoder layer. This space is responsible for receiving the output of the encoder layer, which in the image case is a compressed down-sampled image, and extracting features from this compressed image after the model has been trained over the data set. After being compressed through the encoder layer, the noisy image is delivered to the latent space layer. At this time, The latent space is responsible for receiving the output of the encoder layer which in the image case a compressed down sampled image and then extract features from this compressed image and after the model has been trained over the data set the latent space layer plays the role in generating data on the upcoming noisy image and can be considered as after the code is being trained the noisy image after being compressed through the encoder layer and then delivered to the latent space layer the latent space can now detect the noisy and replace the noisy pixel in the image with a suitable pixel value from the trained data. The most crucial factor in this process is the loss function which measures the MSE between the input image and the reconstructed output. It is a critical parameter to be calculated as the autoencoder model adjusts its weights to decrease the loss function.

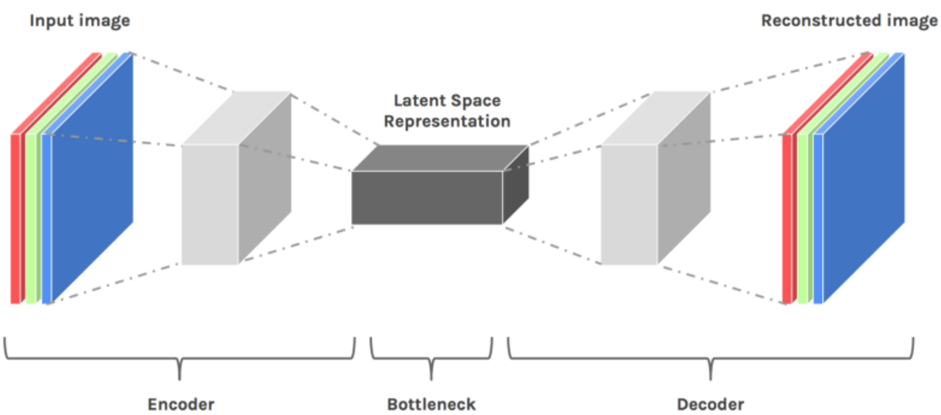

Fig. 6. The architecture of the adopted autoencoder based image denoiser

Although the uncomplicated architecture that the autoencoder model has, it has a significant critical point as its efficiency has an inverse relationship with the image size. In contrast, computational processing has a direct relationship with image size. Since as the image size increases, the computational processing increases because it requires more memory usage to save the trained data, and also, an image with a big size requires more parameters to compress and extract the features. Hence, the autoencoder shows excellent efficiency with a data set generated by Microsoft named MNIST, a considerable dataset containing handwritten numbers. The number of images in this dataset is around 72,000, with a size of 28x28 for each. This dataset has many advantages in testing the autoencoder because it contains many images, so the model can be trained on many images, increasing its accuracy. Also, it consists of small-sized images, making it easy to store in memory and be processed quickly.

On the other hand, another trial uses the FER-13 dataset, which contains around 35,000 images sized in 50x50 images for different people. This data set is used to detect human emotions. The conclusion of applying it through the autoencoder is that the images are not denoised as efficiently as the MNIST dataset. This performance is a result of two main reasons; the first one is that the FER-13 dataset contains half the amount of images the MNIST dataset has. The second reason is that it is larger in image size, requiring additional layers to extract more features and more time and memory to process it. This technique requires 44min to be trained using the MNIST dataset. The results using the MNIST dataset for two different examples are illustrated in Figure. 7.

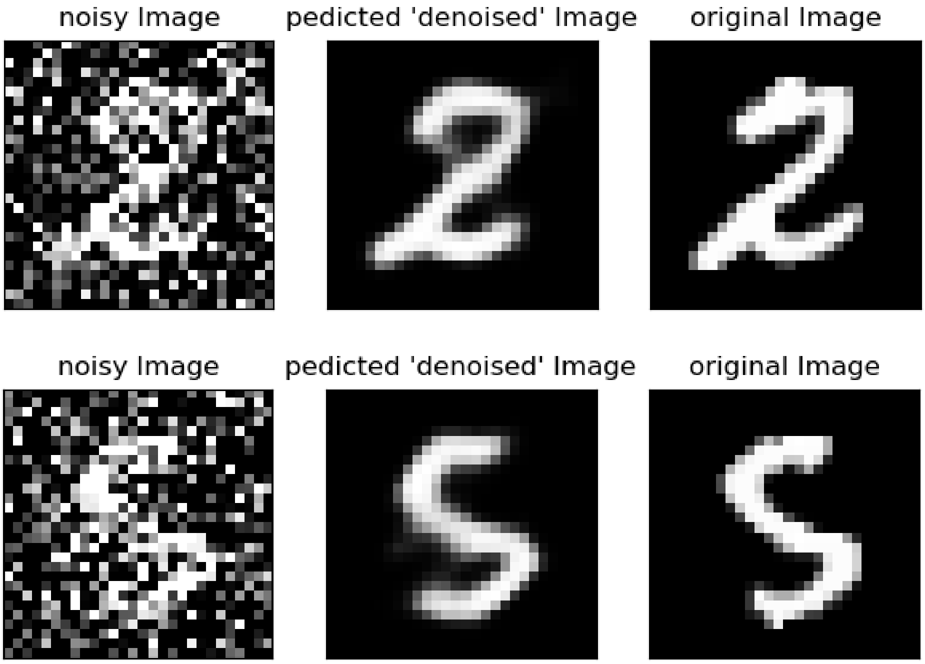

Fig. 6. The results of applying the autoencoder-based proposed technique to two examples of MNIST dataset

The autoencoder was trained on a 60% noisy image resulting in PSNR and SSIM equal to 17.1 dB and 81.9%, respectively. On the other side, when using the autoencoder with the FER-13 dataset that contains 35,000 of 48x48 human face reaction images, the autoencoder layers were increased from 3 layers to 5 layers in each of the encoder and decoder stages the time to train the model is around 25 minutes. At the same time, the efficiency decreased significantly at 60% noise density, as PSNR reached 16 dB. The values of the measured metrics are tabulated in the Table. 3. The results of using the FER-13 dataset for two different examples are illustrated in Figure. 8.

From the aforementioned discussion, this technique is adopted for hardware implementation of the image denoiser for being used in the processing of the images of number which ensures high performance with the impact of the speeding the processing up in case of the hardware implementation rather than the software processing.

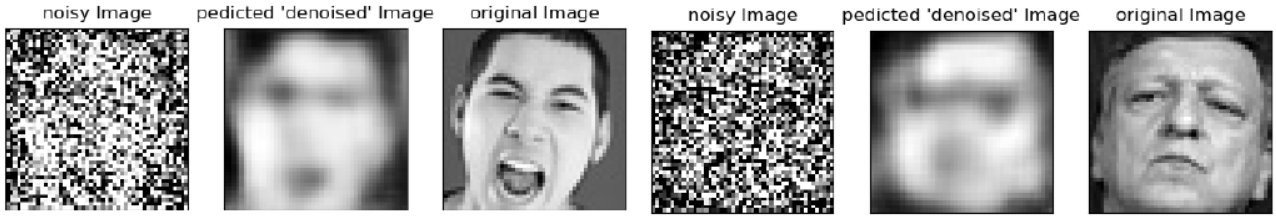

Fig. 8. The results of applying the autoencoder-based proposed technique to two examples of FER-13 dataset

Table 3
The calculating PSNR for the autoencoder proposed based denoiser at 60% noise densities

Number of layers	Dataset	PSNR (dB)
3	MNIST	17.1
5	FER-13	16

From the tabulated results, it is noted that the simple 3-layers autoencoder-based denoiser has superior performance with respect to the most of the standard filters with small degradation about the MDBUTMF algorithms. Notwithstanding the fact that the PSNR in case of the Autoencoder based denoiser is smaller than its value in case of the other AI based techniques but this value is adequate for the applications that involve processing and analyzing images of numbers. Furthermore, it is foreseen that this autoencoder denoiser has high potential to be hardware realized than the other techniques due to its simplicity and low computational cost.

4. Conclusions

Several applications involve processing and analyzing images of numbers, especially in the Internet of Things age. However, the corruption of images by noise is a crucial problem. This reflects the necessity of filtering the noisy images to retrieve their clean versions. There is no perfect filter or technique to denoise all the image types, whether traditional or modern. The denoiser should guarantee sufficient performance and high processing speed for the image of numbers. Hardware implementation can speed up any algorithm instead of using general-purpose processors. Therefore, this paper suggests the autoencoder-based denoiser for hardware implementation for the images of numbers. Two versions of the autoencoder-based denoiser, the three and five layers, are studied. Their performance efficiency versus the complexity is compared to the standard denoising filters and the AI-based denoisers. The results show adequate performance in the case of images of numbers that do not contain many details to be detected with apparent simplicity that is encouragable for being hardware implemented.

Acknowledgement
This research was not funded by any grant.

References

[1] Alisha, P B, and Gnana Sheela K. “Image Denoising Techniques-An Overview” 11, no. 1 (2016): 78–84. https://doi.org/10.9790/2834-11117884.

[2] Bailey, Donald G. *Design for embedded image processing on FPGAs*. John Wiley & Sons, 2023.

[3] Bajaj, Komal, Dushyant Kumar Singh, and Mohd Aquib Ansari. "Autoencoders based deep learner for image denoising." *Procedia Computer Science* 171 (2020): 1535-1541.https://doi.org/https://doi.org/10.1016/j.procs.2020.04.164.

[4] Bindal, Nishant, and Bharat Garg. "Novel three stage range sensitive filter for denoising high density salt & pepper noise." *Multimedia Tools and Applications* 81, no. 15 (2022): 21279-21294. https://doi.org/10.1007/s11042-022-12574-z.

[5] Draz, H. H., N. E. Elashker, and Mervat MA Mahmoud. "Optimized algorithms and hardware implementation of median filter for image processing." *Circuits, Systems, and Signal Processing* 42, no. 9 (2023): 5545-5558. https://doi.org/10.1007/s00034-023-02370-x.

[6] Fu, Bo, Xiaoyang Zhao, Yi Li, Xianghai Wang, and Yonggong Ren. "A convolutional neural networks denoising approach for salt and pepper noise." *Multimedia Tools and Applications* 78, no. 21 (2019): 30707-30721. https://doi.org/10.1007/s11042-018-6521-4.

[7] Garg, Bharat. "Restoration of highly salt-and-pepper-noise-corrupted images using novel adaptive trimmed median filter." *Signal, Image and Video Processing* 14, no. 8 (2020): 1555-1563. https://doi.org/10.1007/s11760-020-01695-3.

[8] Ghumaan, Rajanbir Singh, Prateek Jeet Singh Sohi, Nikhil Sharma, and Bharat Garg. "A novel hybrid decision-based filter and universal edge-based logical smoothingadd-on to remove impulsive noise." *Turkish Journal of Electrical Engineering and Computer Sciences* 29, no. 4 (2021): 1944-1963.. https://doi.org/10.3906/ELK-2005-12.

[9] Hanumantharaju, M. C., M. Ravishankar, D. R. Rameshbabu, and S. Ramachandran. "Adaptive color image enhancement based geometric mean filter." In *Proceedings of the 2011 International Conference on Communication, Computing & Security*, pp. 403-408. 2011. https://doi.org/10.1145/1947940.1948024.

[10] Krestinskaya, O., B. Choubey, and A. P. James. "Memristive GAN in analog." *Scientific reports* 10, no. 1 (2020): 5838. https://doi.org/10.1038/s41598-020-62676-7.

[11] Kulkarni, Rutuja Nandkumar, and Pradip C. Bhaskar. "Decision based median filter algorithm using resource optimized FPGA to extract impulse noise." *International Journal of Reconfigurable and Embedded Systems* 3, no. 1 (2014): 1. https://doi.org/10.11591/ijres.v3.i1.pp1-10.

[12] Lehtinen, Jaakko, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. "Noise2Noise: Learning image restoration without clean data." *arXiv preprint arXiv:1803.04189* (2018).

[13] Li, Ji-Peng Olivia, Hanruo Liu, Darren S J Ting, Sohee Jeon, R V Paul Chan, Judy E Kim, Dawn A Sim, et al. "Gan2Gan." *Progress in Retinal and Eye Research* 82 (2021): 100900. https://doi.org/https://doi.org/10.1016/j.preteyeres.2020.100900.

[14] Liang, Luming, Seng Deng, Lionel Gueguen, Mingqiang Wei, Xinming Wu, and Jing Qin. "Convolutional neural network with median layers for denoising salt-and-pepper contaminations." *Neurocomputing* 442 (2021): 26-35. https://doi.org/10.1016/j.neucom.2021.02.010.

[15] Mafi, Mehdi, Hoda Rajaei, Mercedes Cabrerizo, and Malek Adjouadi. "A robust edge detection approach in the presence of high impulse noise intensity through switching adaptive median and fixed weighted mean filtering." *IEEE Transactions on Image Processing* 27, no. 11 (2018): 5475-5490. https://doi.org/10.1109/TIP.2018.2857448.

[16] Mahmoud, Mervat MA, Nahla E. Elashkar, and Heba H. Draz. "Low-energy Pipelined Hardware Design for Approximate Medium Filter." *ACM Transactions on Design Automation of Electronic Systems* 28, no. 3 (2023): 1-21. https://doi.org/10.1145/3582005.

[17] Örnhag, Marcus Valtonen, Püren Güler, Dmitry Knyaginin, and Mattias Borg. "Accelerating AI using next-generation hardware: Possibilities and challenges with analog in-memory computing." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 488-496. 2023.

[18] Satti, Piyush, Nikhil Sharma, and Bharat Garg. "Min-max average pooling based filter for impulse noise removal." *IEEE Signal Processing Letters* 27 (2020): 1475-1479. https://doi.org/10.1109/LSP.2020.3016868.

[19] Tondare, Santosh M., and Niteen S. Tekale. "An efficient algorithm for removal of impulse noise in color image through efficient modified decision based unsymmetric trimmed median filter." In *2015 IEEE International Advance Computing Conference (IACC)*, pp. 476-480. IEEE, 2015. https://doi.org/10.1109/IADCC.2015.7154754.

[20] Cha, Sungmin, Taeeon Park, Byeongjoon Kim, Jongduk Baek, and Taesup Moon. "GAN2GAN: Generative noise learning for blind denoising with single noisy images." *arXiv preprint arXiv:1905.10488* (2019).