

Article

Dual Autoencoder Network with Separable Convolutional Layers for Denoising and Deblurring Images

Elena Solovyeva *  and Ali Abdullah

Department of Electrical Engineering Theory, Saint Petersburg Electrotechnical University "LETI",
197022 St. Petersburg, Russia

* Correspondence: ebsoloveva@etu.ru

Abstract: A dual autoencoder employing separable convolutional layers for image denoising and deblurring is represented. Combining two autoencoders is presented to gain higher accuracy and simultaneously reduce the complexity of neural network parameters by using separable convolutional layers. In the proposed structure of the dual autoencoder, the first autoencoder aims to denoise the image, while the second one aims to enhance the quality of the denoised image. The research includes Gaussian noise (Gaussian blur), Poisson noise, speckle noise, and random impulse noise. The advantages of the proposed neural network are the number reduction in the trainable parameters and the increase in the similarity between the denoised or deblurred image and the original one. The similarity is increased by decreasing the main square error and increasing the structural similarity index. The advantages of a dual autoencoder network with separable convolutional layers are demonstrated by a comparison of the proposed network with a convolutional autoencoder and dual convolutional autoencoder.

Keywords: machine learning; image processing; computer vision; image denoising; autoencoder; dual autoencoder; convolutional neural network; separable convolutional neural network; deep learning; non-linear model



Citation: Solovyeva, E.; Abdullah, A. Dual Autoencoder Network with Separable Convolutional Layers for Denoising and Deblurring Images. *J. Imaging* **2022**, *8*, 250. <https://doi.org/10.3390/jimaging8090250>

Academic Editor: Pier Luigi Mazzeo

Received: 11 July 2022

Accepted: 5 September 2022

Published: 13 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

After the invention of the camera in 1816, images became a significant information-saving and data-saving unit in our lives. The use of images influenced all the fields of human knowledge, from history and art to medicine and technology and many others. However, one of the main disadvantages of images in physical and digital forms is their lack of resistance to noise and impact. In addition, with the development of digital cameras, new types of noises that can affect an image appeared; therefore, noise reduction in image processing and computer vision started to get more attention from researchers due to its importance, and image denoising became a significant area of research. Many models are built by means of algorithms that identify connections among picture pixels and generate predictions about the value of an individual pixel based on the attributes of the image [1]. Image denoising is applied in the following:

- In image restoration, which is the methodology of reaching the original image from a distorted noisy image. Image restoration is accomplished by inverting the process that blurs or destroys the parts of the original image [2–7].
- In visual tracking, which is the process of tracking objects, animals, or humans using cameras, the tracking environment could have a lot of noise sources caused by natural reasons such as weather like heat or by stimulated reasons such as radiation in nuclear stations [8,9].
- In image registration, which is the technique of overlaying multiple images to be examined, data from several images have to be combined to gather information using different sensors and those could be exploited to noise [10,11].

- In image segmentation, in which noise in pixels can appear when dividing a single digital image into multiple portions called image segments or image regions to facilitate or modify the image presentation into a form, which is more significant and more comfortable to investigate and analyze [12,13].

Significantly, artificial intelligence and machine learning found origins in image denoising because images are one of the main sources of information that can arise from different origins, including the internet, medical archives, historian archives, social media, reviews, etc. Since images are not resilient to noise, understanding information from distorted or noised images would be challenging and difficult, yet after the development of deep learning, image denoising is developing and allowing us to be able to restore the original images and to take a piece of meaningful information from it with minimal effort. Many algorithms and approaches are used to achieve image denoising, such as:

1. Spatial domain filtering is a method for extracting noise by calculating the gray value for the current pixel, relying on itself and the surrounding pixels in the original picture. In general, linear and non-linear filters are the two primary classes of filters [5–7,14–16]. Linear filters in the spatial domain removed noise, but they failed to maintain image consistencies. For example, the Wiener filter was used for Gaussian noise (Gaussian blur or Gaussian smoothing) reduction to overcome the disadvantage of the mean filter because it can overfit the smoothness of an image with high noise [17,18]. However, the Wiener filter sometimes caused the sharp edges in images to be blurred. Non-linear filters, such as the median filter, weighted median filter, and bilateral filter, can reduce noise without any unique identification due to their edge-preserving characteristic [19–21]. This characteristic indicates that the intensity value of each pixel is substituted with a weighted average of the intensity values from nearby pixels rather than a single intensity value.

2. Total variation is a method founded on the principle that the integral of the absolute image gradient increases when the images include excessive details; therefore, it has a high total variation. As a result, reducing a noisy image's overall variance to a near value to the original image removes extraneous detail while maintaining critical elements such as edges. The key benefit of this approach over the previous methods is that it decreases noise while at the same time maintaining the borders of the picture [22–24].

3. Non-local means, in this technique, the main difference from the previous methods is that it depends, unlike them, on all the pixels in the image in calculating the mean. After that, it weights the mean of all the pixels according to their similarity with the target pixel, resulting in significantly better post-filtering clarity and less image information loss [25–27].

4. Data-adaptive transform, which is a technique to improve artifact removal based on the transform domain approach using a blend of data-adaptive and non-data-adaptive transform domains [28]. The removal of artifacts from the scalp electrocardiogram is critical for both automatic and visual examinations of underlying brainwave activity [29].

5. Non-data-adaptive transform is a technique that may be separated into two domains: the spatial-frequency domain and wavelet domain, which is a methodology for transform domain filtering approaches [30]. Low-pass filtering is a spatial-frequency domain filtering technique that works by producing a frequency-domain filter that allows all frequencies below a cut-off frequency to pass through while decreasing all frequencies beyond the cut-off frequency. In contrast, a scale-space representation is created by the use of the wavelet transform [31].

6. A significant area of research is image denoising where many models are built on algorithms that identify connections among picture pixels and generate predictions about the value of an individual pixel based on the attributes of the image. This challenge may be thought of as a supervised learning problem in which each noisy image represents an input, and the intended result is the original shape of the image [32]. This supervised learning model analyzes the observations or input images and provides a solid representation of the connection between the input and output. Research in image processing and computer vision with diverse experiences has started using deep neural networks to solve a vast range

of tasks in these fields, including image classifications, image recognition, face detection, face recognition, emotion recognition, and now image denoising [33–38]. This breakthrough made it possible to use deep neural networks to examine the picture denoising problem. A convolutional neural network (CNN) has the ability to learn noise patterns due to its property of feature extraction; therefore, by constructing a model of multiple layers of CNN, this model will be able to learn these patterns and modify its weights through the training process on big datasets to cancel these noise patterns. Therefore, many researchers in image denoising used convolution neural networks as a base for their models. The main modification for convolution layers is to connect them with other layers such as batch normalization, Relu layers, and Tanh layers to build enhancement blocks, sparse blocks, and restoration blocks in the models, which achieve the denoising process but with a high computational cost. In [39], a feedforward architecture of four stages of filtering to denoise mixed Gaussian-impulse noise was represented. The first stage was a rank order filter formed from existing filters and the adaptive switching weight mean filter, then three stages of the CNN layers for the rest of the filtering process. In [40], two modifications to the use of convolution neural networks were proposed. The first modification combined convolution neural networks, batch normalization, and Relu layers to build a denoise model of four blocks: sparse block, feature enhancement block, attention block, and reconstruction block. The second modification was the feedforward model of five layers of convolution neural networks with Relu layers to get a denoised image and then enhance it using a pretrained semantic segmentation model (SegNet). In [41], a pseudo convolutional neural network model was used to preprocess the image before denoising. The pseudo convolutional neural network model consisted of three layers (a convolution layer, a pooling layer, and a filter layer) to get a preprocessed image. After getting the preprocessed image, the model that combined convolution layers, Relu layers, and batch normalization was used to get the noise from the noisy image and then subtract it from the preprocessed image to get the denoised image.

7. Many researchers have lately started to use autoencoders built with convolutional neural networks (CNNs) to reduce the noise in images [42–46]. Deep neural networks are increasingly being used to handle a wide range of problems in image processing and computer vision research, including picture classifications, image identification, face detection, face recognition, emotion recognition, and even image denoising. This breakthrough made it possible to examine the topic of picture denoising using deep neural networks. To minimize picture noise, several researchers have recently begun to employ autoencoders constructed using convolutional neural networks. Autoencoders with convolution layers were developed from the previous use of autoencoders with fully connected layers. Since this development, the main modifications for using the autoencoder are to connect its output with other outputs from blocks of convolution neural networks or other types of neural networks, then sum these different outputs to get the final output to enhance the output of the autoencoder. In [42], two models were presented to achieve the denoise process. The first model was a combination of convolution neural layers with max-pooling followed by multiple layers of fully connected layers. The second model was an autoencoder with convolution neural networks and pooling layers, where their results showed the benefits of using an autoencoder in image denoising over convolution neural networks and fully connected layers without autoencoders. In [43], three separated structures were suggested that combine autoencoders with feedforward convolution neural networks, pooling layers, and fully connected layers. Each structure was trained independently, and the final output was a sum of the three outputs of each structure. In [44], two structures of deep neural networks were proposed; the first structure was convolution layers to extract global and local noise. In comparison, the second structure was an autoencoder with pixel up-sampling convolution layers to remove the noise features from the image. This dual structure was used to denoise hyperspectral images and work on wavelet transformed bands. In [45], the system consisted of four blocks: a preprocessing block, preparing synthetic image block, stacked convolutional autoencoder block, and validation measure block. Therefore, after the image

was preprocessed and prepared, it entered the stacked convolutional autoencoder (SCAE) block, where the learning process took place to remove the noise, and then it was validated in the validation measure block. In [46], the denoising system consisted of two stages. The first stage was the fast region-based convolutional neural networks (fast R-CNNs) to restrict the areas in the images that contained the noise. The second stage was denoising the images using a model of shared layers of convolution neural networks followed by two structures of deep denoised autoencoder (DDAE) and deep material classifier (DMC). The DMC consisted of two convolution layers followed by fully connected layers and shared the first two convolution layers with the autoencoder DDAE, where the DDAE consisted of fully connected layers followed by three deconvolution layers.

In general, these modifications gave the use of the autoencoder the ability to extract features from non-linear processes and perform non-linear transformations image denoising. However, the main disadvantage was an increase in the computational cost and the time of training and performing. In this study, the first modification to the use of an autoencoder is to build a dual autoencoder. The second modification is to use separable convolution neural networks instead of convolution neural networks. These modifications give our model the additional advantages of increasing the accuracy of the image restoration, reducing the training parameters, decreasing the training and performing time of the autoencoder's network, and saving memory.

Thus, we can emphasize the following pros for the proposed methodology. First, we use a special type of neural network, namely, an autoencoder network, which extracts features from non-linear processes and transformations by means of dimensionality reduction when compressing the data and information retrievals when decompressing the data. Second, the dual autoencoder network ensures a higher accuracy of denoising images in comparison with one autoencoder network. Third, the use of separable convolution neural networks in dual autoencoders results in decreasing the number of networks parameters and, consequently, the amounts of the calculation time and occupied memory. The con for methodology, which is characteristic of autoencoders, consists in the difficulty to reach the high image restoration quality under a large number of small features.

The paper contains six sections. The significance of image denoising and its application fields, the main algorithms, and approaches are used to achieve the image denoising mentioned in the introduction. The main noise that can affect an image is described in Section 2. The mathematical model of autoencoder and dual autoencoder using convolutional neural networks and separable convolutional neural networks (SCNNs) is represented in Section 3. The results of image denoising, their comparison, and corresponding conclusions are described in Sections 4–6.

2. Basic Types of Noise on Images

Noise is generally described as an accidental variation in brightness or pixel information, and it is often created by accident or by the limitations of the image sensor or camera. Improper environmental circumstances can also create noise. These problems are repeatedly unavoidable in real systems, which increases the importance of image denoising in image processing and computer vision. Gaussian noise (Gaussian blur), Poisson noise, Speckle noise, and impulse noise are considered the main noises that can affect an image.

2.1. Gaussian Noise

The term “Gaussian noise,” anointed after Carl Friedrich Gauss, refers to a sort of signal noise with a probability density function equivalent to that of the normal distribution (Gaussian distribution) [47], which means the potential values of the noise are dispersed according to the Gaussian distribution depending on Equation (1). The normal distribution formula is given as the following:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (1)$$

where $G(x)$ is the normal distribution function; x is the input pixel value; σ is the standard deviation; and μ is the mean of the values of the input pixels surrounding a current pixel. Sensor noise caused by low light or extreme temperature, as well as transmission noise such as electrical circuit noise, are the main origins of Gaussian noise in digital photographs. Gaussian noise may be minimized using a spatial filter in digital image processing; however, when smoothing a picture, an unwanted byproduct may be the blurring of fine-scaled image edges and features, which correspond to blocked high frequencies [48]. Mean filtering, median filtering, and Gaussian smoothing are three common spatial filtering techniques for noise reduction [49].

2.2. Poisson Noise

The quantized character of light and the independence of photon detections make Poisson noise a primary kind of uncertainty related to light measurement. Except in low-light settings, its predicted magnitude is signal-dependent, and it is the most common cause of picture noise [50]. The discrete character of electric charge causes shot noise in electronics. Shot noise may also be seen in photon counting in optical systems, where it is linked to light's particle nature [51]. Each photon observation can be considered a separate event with a randomized distribution pattern. The discrete probability distribution can define the Poisson process:

$$P(x) = \frac{e^{-\lambda t} (\lambda t)^x}{x!}, \quad (2)$$

where $P(x)$ is the discrete probability distribution; x is the total number of photons detected by a sensor element during a period of time t ; λ is the amount of photons anticipated per unit time period; and t is the time interval.

Equation (2) is a typical Poisson distribution with the anticipated photoelectric effect number as the rate parameter λt . Poisson noise is the uncertainty indicated by this distribution. Typically, this estimate is highly accurate. Poisson noise is usually controlled by other signal-independent forms of noise for lower photon counts. Photon noise is a low constraint on the uncertainty of measuring light since it is generated from the nature of the signal. Any test would be prone to Poisson noise even under perfect imaging circumstances, devoid of all other sensor-based causes of noise [52]. Imaging is considered photon-limited when Poisson noise is the only substantial source of uncertainty, which is frequent in bright photon-rich situations.

2.3. Speckle Noise

Speckle is a particulate disturbance that occurs naturally in radars, synthetic aperture radar, medical ultrasonography, and optical coherent tomography pictures and decreases their clarity. On the frequency scale, most artificial and natural surfaces are highly harsh. The common interference phenomena of speckle affect images acquired from these surfaces by coherent imaging methods such as laser, synthetic aperture radar, and ultrasound. If we describe our reflectivity function as an array of scatterers, we can understand where this behavior comes from. Because of the finite resolution, it is transmitted at any given moment from a large number of potential distributions inside the resolution cell. These dispersed signals add coherency. Depending on the relative phases of each distributed signal, they add positively or harmfully. These constructive and destructive interference patterns, depicted as bright and dark spots in the picture, cause speckles. Whenever the diffuser's radiation pattern is considered the primary important parameter for producing an image during the imaging process, and the change in Radom light phases during the reflection process is not taken into account, speckles in pictures can be treated completely as noise [53,54]. The phase function of diffused lights may be thought of as a non-correlation random field with an equal distribution between 0 and π where the optically surface may be thought of as a collection of several random diffused patches described in Equation (3). Evaluating the features of a true speckle pattern reveals that it invariably comprises signal-

dependent disturbances such as speckle noise. The speckle pattern mathematical model can be described as the following [55]:

$$g(x, y) = f(x, y) + r[f(x, y)] \times [n_1(x, y) + n_2(x, y)] + n_3(x, y) + p(x, y), \quad (3)$$

where $g(x, y)$ is the speckle pattern; x, y is the dimension of the image; $f(x, n)$ is the intensity of the original image; $n_2(x, y)$ and $n_3(x, y)$ are statistically independent and signal-independent from one another, random Gauss distribution noise with zero mean; $n_1(x, y)$ is a random noise that has some spatial correlation, the degree of which is directly related to the amplitude of the imaging system's incoherent point spread function; $p(x, y)$ is the pulse noise whose amplitude probabilistic density is evenly distributed over the image's dynamic range; and $r[f(x, y)]$ is the original image pattern's compound function.

2.4. Impulse Noise

Salt and pepper noise is another name for impulse noise where sharp and rapid disruptions in the visual signal might create this noise. It manifests as a scattering of white and black pixels [5–7,56–58]. Transmitting data faults, memory cell failures, and analog-to-digital converter flaws can all produce this issue. It takes the shape of white and black pixels that appear at random and can drastically degrade image quality. Noisy pixels in photos degraded by salt and pepper noise are alternately adjusted to the minimum and highest intensity values, giving the image a “salt and pepper” look described in Equation (4). Unaffected pixels, however, are never affected [59,60]. The following mathematical equation may be used to express the impulse noise:

$$F(x, y) = \begin{cases} s_{\min} & \text{with probability } p \\ s_{\max} & \text{with probability } q \\ I(x, y) & \text{with probability } 1 - p - q \end{cases}, \quad (4)$$

where $F(x, y)$ is the noisy image matrix with impulse noise with dimensions (x, y) ; s_{\min} is the minimum value of the pixels in the original image $I(x, y)$; s_{\max} is the maximum value of the pixels in the original image $I(x, y)$; and $I(x, y)$ is the original image matrix without the impulse noise.

3. Dual Autoencoder with Convolutional and Separable Convolutional Neural Networks

Autoencoders are a kind of artificial neural network design utilized to develop efficient coding for unlabeled input. In the unsupervised learning process, the architecture of neural networks is such that we may place a bottleneck in the network, which pushes the network to represent the original input in a compressed manner [61]. Assuming that each of the input attributes is completely independent of the others, this compression and reconstruction operation will be quite challenging. In the case of some structure in the data such as the correlations between input characteristics, this structure may be learned and used to the advantage of the user when driving the input through the bottleneck in the network. Autoencoders are primarily dimensionality reduction (or compression) algorithms with several critical characteristics, including the following:

1- They are data-specific. Autoencoders can only significantly compress data similar to the data on which they were previously trained. Because they learn characteristics particular to the training data, they differ from a basic data compression technique such as zip in that they learn features specific to the training data [62]. This means that a landscape photo cannot be compressed by an autoencoder that has been trained on handwritten numbers.

2- They are loosely encoding. The output of the autoencoder will not be an exact replica of the input; rather, it will be a close but degraded copy of the original [63].

3- They are an unsupervised learning technique. Autoencoders are classified as an unsupervised learning technique because they do not require explicit labels to be trained

on to be effective [60]. To be more specific, they are self-supervised in the sense that they use the training data they are given to make their own labels.

In order to construct an autoencoder, three components are required: an encoding method, a decoding method, and a loss function for comparing the output to the target signal.

The autoencoder model achieves a balance between the properties and characteristics listed previously. To properly rebuild the scene, it must pay close attention to the input signals. A bottleneck (in Figure 1) restricts the amount of information that can be transferred throughout the whole network due to congestion. By punishing the network depending on the reconstruction error, our model may learn how to restore the original input from an “encoded” state in the most efficient manner feasible [64]. The autoencoder structure is shown in Figure 1 and the model is described as Equation (5):

$$\begin{aligned} \Phi : X + N &\rightarrow F \\ \Psi : F &\rightarrow X' \\ Loss &= \underset{\Phi, \Psi}{\operatorname{argmin}} \|X - Y\| \end{aligned} \quad (5)$$

where Φ is the encoder function that maps the noisy image into the space F that represents the bottleneck; X is the matrix of the original image; N is the noise matrix added to the original image; F describes the space of matrices in the bottleneck; Ψ is the decoder function that maps the space F of the bottleneck into the output; X' is the output matrix of the autoencoder; and $Loss$ represents the loss function that aims at minimizing the error between the desired output X and the current output X' .

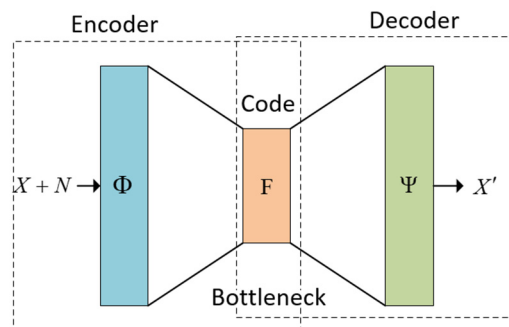


Figure 1. The autoencoder structure.

Based on the autoencoder structure, a dual autoencoder is proposed to denoise images. The dual autoencoder structure is shown in Figure 2.

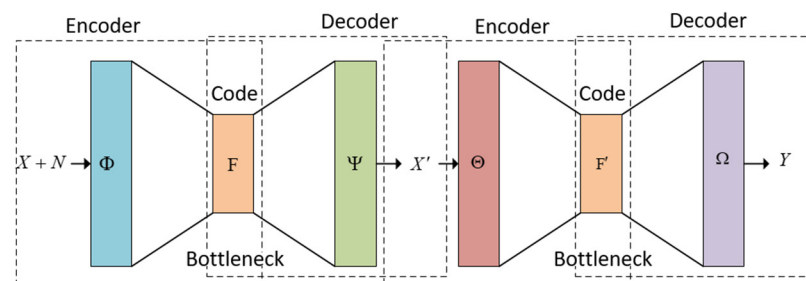


Figure 2. The dual autoencoder structure.

As can be seen from Figure 2, there are four kinds of mapping: two for denoising the image and two for enhancing the quality of the denoised image. The model of the dual autoencoder is described as the following:

$$\begin{aligned}
 &\Phi : X + N \rightarrow F \\
 &\Psi : F \rightarrow X' \\
 &\Theta : X' \rightarrow F' \\
 &\Omega : F' \rightarrow Y \\
 &Loss = \underset{\Phi, \Psi, \Theta, \Omega}{\operatorname{argmin}} \|X - Y\|
 \end{aligned} \tag{6}$$

where X is the matrix of the original image; N is the noise matrix added to the original image; X' is the output matrix of the first autoencoder; Y is the output matrix of the second autoencoder; Φ and Θ are the encoder functions of the first and second autoencoders, which map the matrix $(X + N)$ into the space F of matrices and the matrix X' into the space F' , correspondingly; Ψ and Ω are the decoder functions of the first and second autoencoders, which map the space F into the matrix X' and the space F' into the output Y , accordingly; and $Loss$ represents the loss function that aims at minimizing the error between the desired output X and the current output Y .

In the case of deep autoencoders, we must additionally consider the capabilities of our encoder and decoder models, respectively. Ensuring that the autoencoder model is not just learning an effective technique to remember the training data is critical. Otherwise, the model will fail. Similarly, to supervised learning concerns, we may use various regularization techniques in the network to promote excellent generalization properties. Because of the input data, a sparse autoencoder is constrained to activate sections of the network selectively. In contrast, a complete autoencoder is forced to employ the whole network for every observation. This implies that we may pick a latent state representation (i.e., encoding dimensionality) that makes sense in the data context while still enforcing regularization via the sparsity restriction on the latent state representation. One consequence is that we can allow our network to sensitize certain hidden layer nodes to specific properties of the raw data as a result [65].

Autoencoders can be built from different kinds of neural networks, but since the aim of this research is image denoise, they were built using separable convolutional neural networks and later with a convolutional neural network (see Figures 3 and 4). The neural architecture of the autoencoder depicted in Figure 3 corresponds to the structure in Figure 1, and the neural architecture of the dual autoencoder in Figure 4 corresponds to the structure in Figure 2.

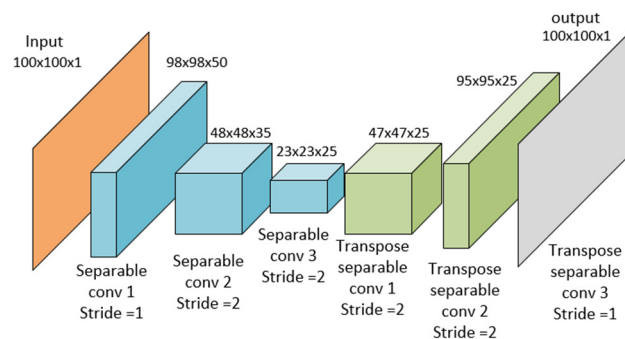


Figure 3. The neural architecture of the autoencoder.

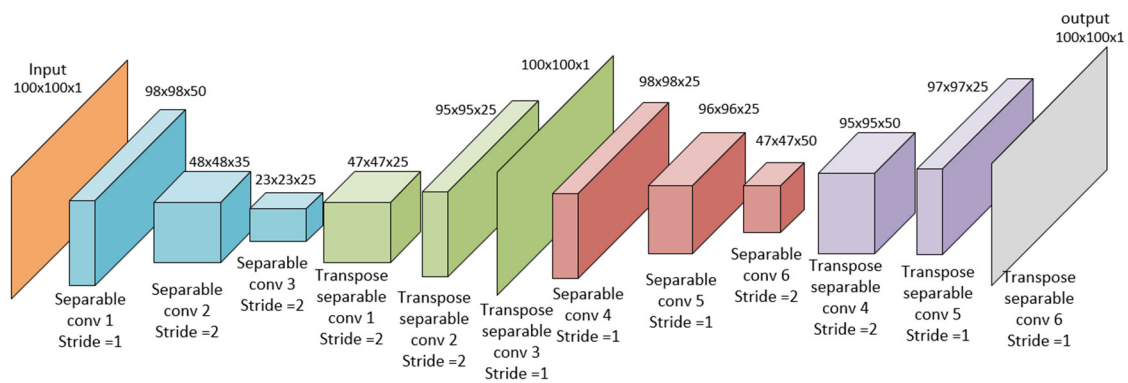


Figure 4. The neural architecture of the dual autoencoder.

As shown in Figures 3 and 4, the number of kernels in the last layer gives the ability of this model of autoencoder to denoise colored images or gray images. In our experiment, we used a transpose convolution layer with $100 \times 100 \times 1$; therefore, the denoise process is on gray scale images. However, when changing the parameters in the last layer to three kernels instead of one, the same model will be able to denoise colored images. The noisy image of size $100 \times 100 \times 1$ is the input of the autoencoder illustrated in orange. Then, the encoding presented in blue takes place by reducing the input size 100×100 to 23×23 . The encoder marked in blue comprises three layers, and every of them is a separable convolutional neural network. These networks have filters $3 \times 3 \times 50$, $3 \times 3 \times 35$, and $3 \times 3 \times 25$ with 50, 35, and 25 kernels, correspondingly. Then, the decoding process is shown in green; this process raises the output of each layer to reach the $100 \times 100 \times 1$ image similar to the input one but with noise reduction. Like Figure 2, the autoencoder in Figure 3 is added to another autoencoder with the same architecture but with the softer size decreased to enhance and denoise. The encoding process is described in Equations (7)–(10), whereas the decoding process is described in Equations (11)–(14) for using both convolutional and separable convolutional neural networks. The purpose of building the autoencoder and dual autoencoder using separable convolutional neural networks and convolutional neural networks is to compare between the performance of the two types of neural network knowing that separable convolutional neural networks achieve a convolution product with a smaller number of learnable parameters. The capability of the convolution layer to extract data with local characteristics is directly connected to the complexity of this layer. Because of its consistent structure, it is more analogous to the neural networks seen in the human brain. It also makes the model easier to understand because it reduces the total number of weights. Padding is added to the input matrix right before the product is finished so that it can accommodate all of the inputs. Padding parameters can have two values, zero or one. The primary use of the padding parameter is to resolve how many elements to add to the original image or matrix to increase its size in a way that makes all its pixels reachable from all edges. This allows the output to have a smaller size or a bigger size with the deconvolution product. The “stride” is the step taken by the kernel or the filter to make the product. The term “stride” refers to the stage in the convolution product process. When performing a convolution layer on two matrices, the kernel or filter is used to operate and update the learnable parameters for the layer. Separable convolution layers, however, are fully associated with dimensions since the kernel matrix is split into a product between a row and a column that produces this matrix, which reduces the trainable parameters in such a way that a kernel matrix \mathbf{K} with the size of $(f \times f)$ can be represented with two partial kernels \mathbf{K}_1 and \mathbf{K}_2 with the size of $(f \times 1)$ and $(1 \times f)$. In this way, for example, the learnable parameters of a kernel with size 3×3 will be reduced to six parameters instead

of nine [66]. The main equation that describes the convolution product of the ordinary convolution layer is described in the following equation:

$$\text{Conv}(\mathbf{X}, \mathbf{K}) = \varphi\left(\sum_{i=1}^{n_h} \sum_{j=1}^{n_w} \mathbf{X}_{i,j} \mathbf{K} + b\right), \tag{7}$$

$$\text{Dim}(\text{Conv}(\mathbf{X}, \mathbf{K})) = \left(\frac{n_h + 2p - f}{s} + 1, \frac{n_w + 2p - f}{s} + 1\right), \tag{8}$$

where $\text{Conv}(\mathbf{X}, \mathbf{K})$ represents the convolution product of matrix \mathbf{X} of size $(n_h \times n_w)$ with a kernel matrix \mathbf{K} of size $(f \times f)$; φ is the sigmoid activation function; i and j are counters; $\mathbf{X}_{i,j}$ is a sub-matrix of size $(f \times f)$ of the input matrix with its padding; b is the bias; $\text{Dim}(\text{Conv}(\mathbf{X}, \mathbf{K}))$ is the dimension of the results of the convolution product; n_h and n_w are the elements of the matrix \mathbf{X} dimension; p is the padding parameter; s is the number of strides; and f is the kernel size.

When using the separable convolution, Equation (7) is represented in two convolution products: one is with the partial kernel \mathbf{K}_1 of size $(f \times 1)$ and the second is with the partial kernel \mathbf{K}_2 of size $(1 \times f)$. In this way, the computational cost represented by the learning parameters and the time of learning will be reduced while getting the same and sometimes enhanced results. The equation describing the separable convolution product is described as the following:

$$\bar{\mathbf{X}} = \text{Conv}(\mathbf{X}, \mathbf{K}_1) = \sum_{i=1}^{n_h} \sum_{j=1}^{n_w} \mathbf{X}_{i,j} \mathbf{K}_1, \tag{9}$$

$$\text{Conv}(\bar{\mathbf{X}}, \mathbf{K}_2) = \sum_{i=1}^{(n_h+2p-f)/s+1} \sum_{j=1}^{n_w} \mathbf{K}_2 \bar{\mathbf{X}}_{i,j}, \tag{10}$$

where $\bar{\mathbf{X}}$ is the output of the convolution product $\text{Conv}(\mathbf{X}, \mathbf{K}_1)$ of the input matrix \mathbf{X} with the kernel \mathbf{K}_1 of size $(f \times 1)$, giving a matrix of size $((n_h + 2p - f)/s + 1) \times n_w$; p is the padding parameter; s is the number of strides; $\mathbf{X}_{i,j}$ is a column from the input matrix in the position (i, j) ; $\text{Conv}(\bar{\mathbf{X}}, \mathbf{K}_2)$ is the convolution product between matrix $\bar{\mathbf{X}}$ and the kernel \mathbf{K}_2 of size $(1 \times f)$; and $\bar{\mathbf{X}}_{i,j}$ is a row from the matrix $\bar{\mathbf{X}}$ in the position (i, j) .

When we use a separable convolutional neural network, Equation number (7) is represented by Equations (9) and (10). In contrast, Equation (8), which illustrates the size of the output of the convolution product, will stay the same. The encoding process is described with the Equations (7), (9) and (10), whereas the decoding process is understood to be the transpose of the convolution product, which means we took the transpose of Equations (7), (9) and (10) to achieve the decoding process of the autoencoder. Mathematically, instead of multiplying a partial part from the input matrix \mathbf{X} of size $(n_h \times n_w)$ by the \mathbf{K} of size $(f \times f)$ in Equation (7) or by two sub-kernels \mathbf{K}_1 of size $(f \times 1)$ and \mathbf{K}_2 of size $(1 \times f)$ in Equation (9) and (10), we multiply each value in the input matrix of size $(n_h \times n_w)$ by the kernel of size $(f \times f)$ to produce a $(f \times f)$ matrix or by \mathbf{K}_1 and \mathbf{K}_2 to produce a column of size $(f \times 1)$ or a row of size $(1 \times f)$. Then, we combine all the resulting matrices or vectors according to the initial positions in the input layer and add the overlaid values together with an output padding parameter p_o [67]. Therefore, we can describe the transpose of Equation (7) as the following:

$$\text{TransConv}(\mathbf{X}, \mathbf{K}) = \varphi\left(\sum_{i=1}^{n_h} \sum_{j=1}^{n_w} x_{i,j} \mathbf{K} + b\right), \tag{11}$$

$$\text{Dim}(\text{TransConv}(\mathbf{X}, \mathbf{K})) = ((n_h - 1) \times s + f - 2p + p_o, (n_w - 1) \times s + f - 2p + p_o), \tag{12}$$

where $\text{TransConv}(\mathbf{X}, \mathbf{K})$ represents the transpose convolution product of matrix \mathbf{X} of size $(n_h \times n_w)$ with a kernel matrix \mathbf{K} of size $(f \times f)$; φ is the activation function; i and j are coun-

ters; $x_{i,j}$ is the pixel in matrix \mathbf{X} , after we add padding and bias b ; $\text{Dim}(\text{TransConv}(\mathbf{X}, \mathbf{K}))$ is the dimension of the results of the transpose convolution product; n_h and n_w are the elements of the dimension of the matrix \mathbf{X} ; p is the padding parameter; s is the number of strides; p_o is the output padding parameter; and f is the kernel size.

In the same way, the transpose of the separable convolution product in Equations (9) and (10) can be represented as the following:

$$\bar{\mathbf{X}} = \text{TransConv}(\mathbf{X}, \mathbf{K}_1) = \sum_{i=1}^{n_h} \sum_{j=1}^{n_w} x_{i,j} \mathbf{K}_1, \tag{13}$$

$$\text{TransConv}(\bar{\mathbf{X}}, \mathbf{K}_2) = \sum_{i=1}^{(n_h-1) \times s + f - 2p + p_o} \sum_{j=1}^{n_w} \mathbf{K}_2 \bar{x}_{i,j}, \tag{14}$$

where $\bar{\mathbf{X}}$ is the output of the transpose convolution product $\text{TransConv}(\mathbf{X}, \mathbf{K}_1)$ of the input pixel in the position (i, j) in the matrix \mathbf{X} with the kernel \mathbf{K}_1 of size $(f \times 1)$, the output is a matrix of size $((n_h - 1) \times s + f - 2p + p_o) \times n_w$; p is the padding parameter; p_o is the output padding parameter; s is the number of strides; $x_{i,j}$ is the pixel from the input matrix in the position (i, j) ; $\text{Conv}(\bar{\mathbf{X}}, \mathbf{K}_2)$ is the convolution product between matrix $\bar{\mathbf{X}}$ and the kernel \mathbf{K}_2 of size $(1 \times f)$; and $\bar{x}_{i,j}$ is the pixel of matrix $\bar{\mathbf{X}}$ in the position (i, j) .

From Equations (7)–(10), we can understand the encoding process where the size of the input image has been reduced; from Equations (11)–(14), we can understand the decoding process, where the size of the resulting output is increased till it matches the original size of the input image. This process of reducing the size in the encoding allows us to remove pixels from the input noisy image and later restore those pixels in the decoding process in a way that cancels the noise and predicts the original image pixels before adding the different noises.

4. Results of Image Denoising

In our practical experiment, we used Python 3.9.7 and Tensorflow 2.6.0 with a Pycharm.3.1 2019 environment on an Asus ZenBook Core i7 2.8 GHz 32 G RAM.

The used dataset was 99,534 images of size 100×100 from a combination of famous datasets of people’s faces: IMDB-WIKI, FER-2013, KDEF, and AffectNet [68–71]. We chose these datasets because face images easily get affected by different types of noises, and many technologies have started to rely on face recognition and face detection, adding more importance to this type of image in the present and future [72]. Additionally, this combination of the dataset has various races, ages, and genders. In the training process, the combined dataset was divided into 80,000 images for training and the rest for testing. We started by adding the four kinds of noise using the scikit-image library in Python. The amount of Gaussian noise (Gaussian blur) was between medium and high with a standard deviation of 40, Poisson noise was generated from each image value using the scikit-image library, the speckles’ noise amount was 20%, and the amount of salt and pepper noise was 30%. After adding these noises to the combined dataset, we got four datasets, one for each of the previous noise types. These datasets were used as an input to the autoencoder and dual autoencoder, and the outputs of the autoencoder and dual autoencoder were compared with the original dataset without the noise using the structural similarity index, which is known as an estimation to compare the similarity of two images and is described by the following Equation [73]:

$$\text{SSIM}(\mathbf{X}, \mathbf{Y}) = \frac{(2\mu_X\mu_Y + (0.01l)^2) + (2\sigma_{XY} + (0.03l)^2)}{(\mu_X^2 + \mu_Y^2 + (0.01l)^2)(\sigma_X^2 + \sigma_Y^2 + (0.03l)^2)}, \tag{15}$$

where $\text{SSIM}(\mathbf{X}, \mathbf{Y})$ is the structural similarity index between matrix \mathbf{X} and matrix \mathbf{Y} representing two images; μ_X is the average mean of the pixels of image \mathbf{X} ; μ_Y is the average mean

of the pixels of image \mathbf{Y} ; l is the dynamic range of the pixel values; $\sigma_{\mathbf{X}\mathbf{Y}}$ is the covariance of image \mathbf{X} and image \mathbf{Y} ; $\sigma_{\mathbf{X}}$ is the variance of image \mathbf{X} ; and $\sigma_{\mathbf{Y}}$ is the variance of image \mathbf{Y} .

Moreover, to estimate the quality of the image restoration, we used the peak signal-to-noise ratio (PSNR), described by the following equation [74]:

$$\text{PSNR} = 10 \log_{10} \left(\frac{(2^a - 1)^2}{\frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w |\hat{y}_{i,j} - y_{i,j}|^2} \right), \quad (16)$$

where a is the bit per pixel; h is the height of the image; w is the width of the image; i and j are counters; $\hat{y}_{i,j}$ is the pixel value in the position i and j of the original image; and $y_{i,j}$ is the pixel value in the position i and j of the denoised image. In our investigation, Equation (16) includes $a = 8$, $h = 100$, $w = 100$. The higher the PSNR, the lower the distortion level.

In our study, we compared the performance of the suggested model of the dual autoencoder with the performance of one autoencoder; these models were built with separable convolutional neural networks and convolutional neural networks. The aim of comparing the separable convolutional neural network with the convolutional neural network is to prove that the separable convolution performs as good as convolutional neural networks in both a dual autoencoder and one autoencoder, and that researchers could rely on them in this field of study where they give the same results or even better results with less computational cost and less trainable parameters, which saves time and memory. When using separable convolutional neural networks and convolutional neural networks, the autoencoder shown in Figure 3 takes a noisy image as an input of size 100×100 and contains six layers. The size of filters in the layers except the last layer equals 3×3 . The first layer contains 50 kernels of filters and one stride, giving an output of the size $98 \times 98 \times 50$; the second layer consists of 25 kernels and two strides, giving an output with a size of $48 \times 48 \times 25$; the third layer consists of 25 kernels and two strides, giving an output of size $23 \times 23 \times 25$. The fourth layer is a transpose layer with 25 kernels and two strides, giving an output of the size $47 \times 47 \times 25$. The fifth layer is a transpose layer with 25 kernels and two strides, giving an output of size $95 \times 95 \times 25$ and the last layer is a transpose layer with one kernel of the size 6×6 and one stride giving the final output as 100×100 to match the input size. For the use of separable convolutional neural networks and convolutional neural networks, the dual autoencoder shown in Figure 4 takes a noisy image as an input of size 100×100 and consists of twelve layers, six layers for the first encoder and six for the second encoder. The first six layers are described above. The seventh layer consists of 25 kernels with a size of 3×3 with one stride giving an output of $98 \times 98 \times 25$. The eighth layer consists of 25 kernels with a size of 3×3 and one stride, giving an output with a size of $96 \times 96 \times 25$; the ninth layer consists of 50 kernels of the size of 3×3 and two strides, giving an output of size $47 \times 47 \times 50$. The tenth layer is a transpose layer with 50 kernels with a size of 3×3 and two strides, giving an output of the size $95 \times 95 \times 50$. The eleventh layer is a transpose layer with 25 kernels with the size of 3×3 and one stride, giving an output of size $97 \times 97 \times 25$. The last layer is a transpose layer with one kernel of the size 4×4 and one stride, giving the final output as 100×100 .

The learning process was supervised learning, where the input for each model was the training noised dataset, and the targeted output was the original images' training dataset. For each model, the training process consisted of 40 epochs with a batch size of 200, and for the backpropagation in the training process, we used a binary cross entropy loss function, and an Adam optimizer with 0.001 as a learning rate. The evaluation was on the testing dataset, and the denoised output of the model after training was compared with the original images' testing dataset using the structural similarity index (SSIM) and the peak signal-to-noise ratio (PSNR).

We achieved image denoising for the testing dataset using the trained model; then, the images predicted by the model were compared with the original images before adding

noise using the structural similarity index (15). We presented the results as a histogram of the structural similarity index (15) for the testing data, which shows the accuracy of our noise canceling as a percentage of the similarity knowing that the similarity of noisy images with the original images was zero percent., The histograms of similarity for one autoencoder and a dual autoencoder based on a convolutional neural network (Figure 5a,b) and separable convolutional neural network (Figure 6a,b) when canceling Gaussian noise (Gaussian blur) are depicted below. Corresponding histograms of similarity are shown in Figures 7 and 8, when canceling Poisson noise, in Figures 9 and 10, when canceling speckle noise, and in Figures 11 and 12, when canceling impulse noise.

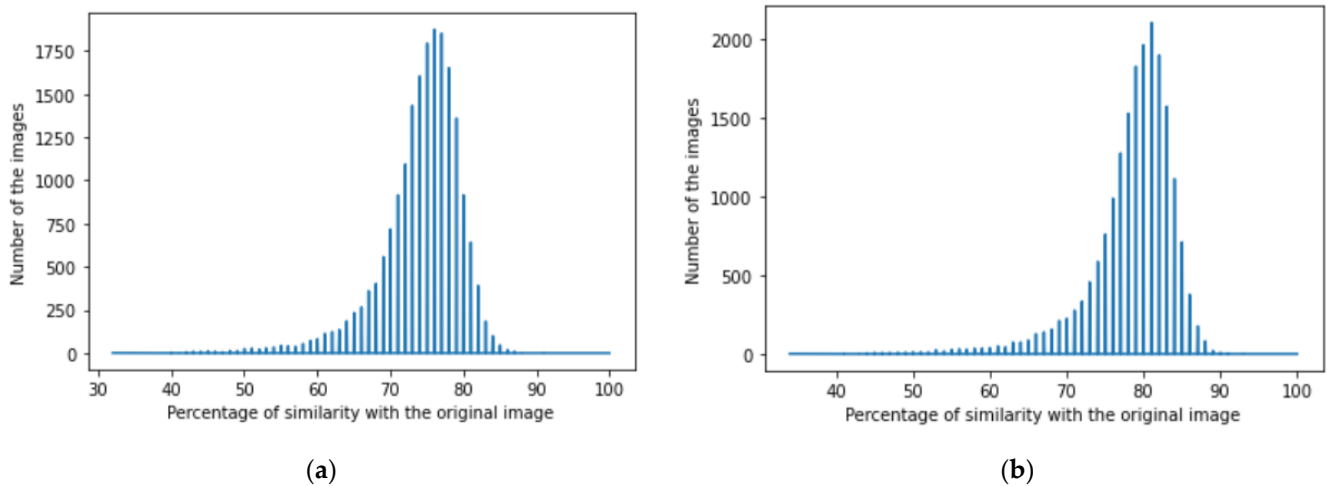


Figure 5. The histograms of similarity between the predicted denoised image and the original image in the case of Gaussian noise: (a) one autoencoder with convolutional neural networks; (b) dual autoencoder with convolutional neural networks.

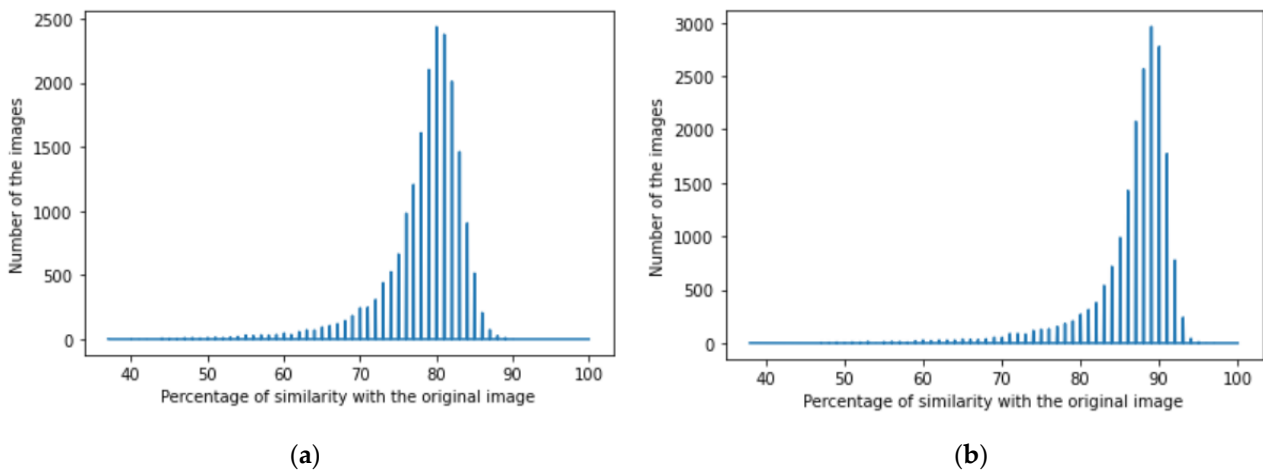
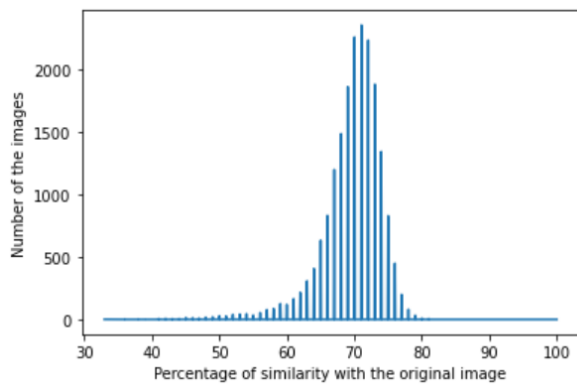
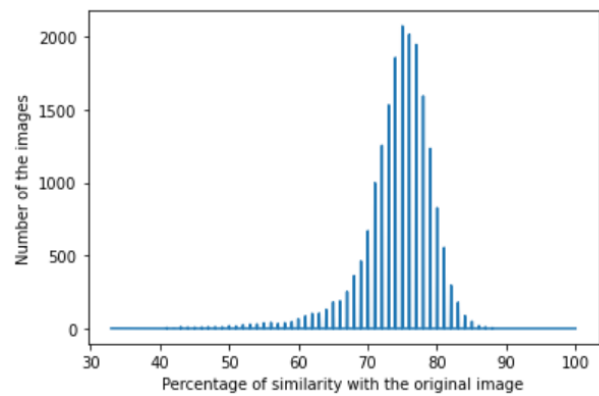


Figure 6. The histogram of similarity between the predicted denoised image and the original image in the case of Gaussian noise: (a) one autoencoder with separable convolutional neural networks; (b) dual autoencoder with separable convolutional neural networks.

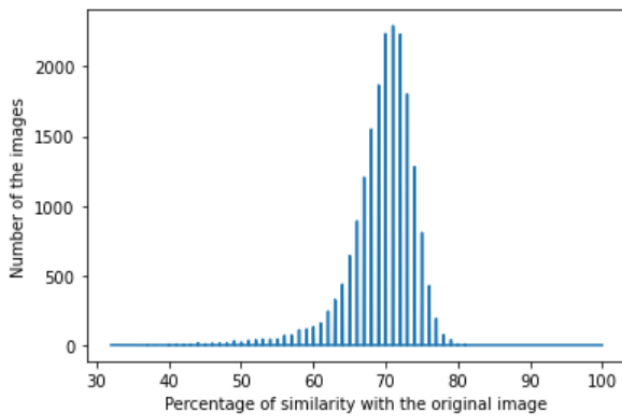


(a)

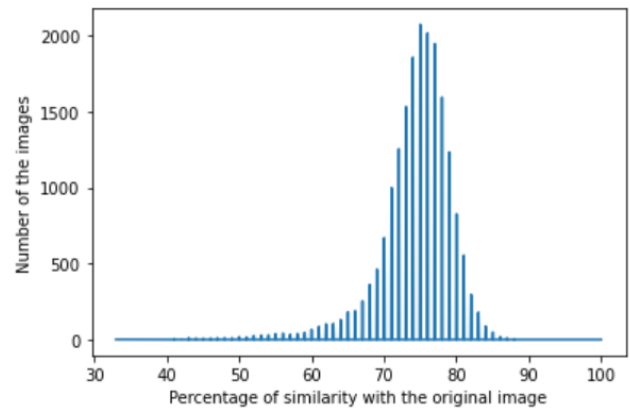


(b)

Figure 7. The histogram of similarity between the predicted denoised image and the original image in the case of Poisson noise: (a) one autoencoder with convolutional neural networks; (b) dual autoencoder with convolutional neural networks.

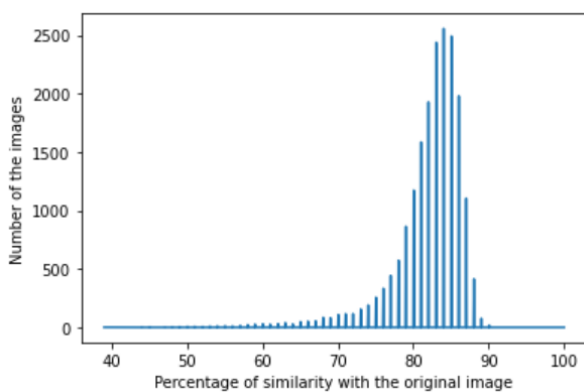


(a)

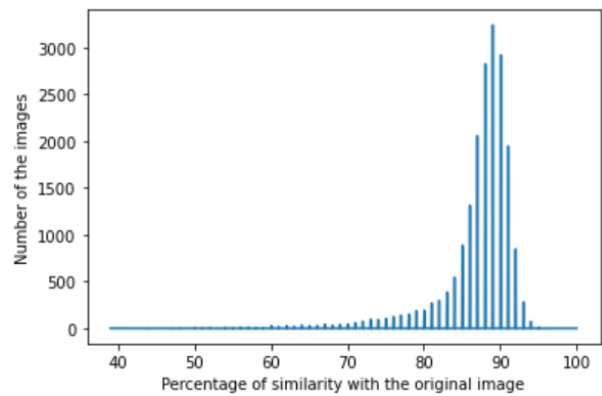


(b)

Figure 8. The histogram of similarity between the predicted denoised image and the original image in the case of Poisson noise: (a) one autoencoder with separable convolutional neural networks; (b) dual autoencoder with separable convolutional neural networks.



(a)



(b)

Figure 9. The histogram of similarity between the predicted denoised image and the original image in the case of speckle noise: (a) one autoencoder with convolutional neural networks; (b) dual autoencoder with convolutional neural networks.

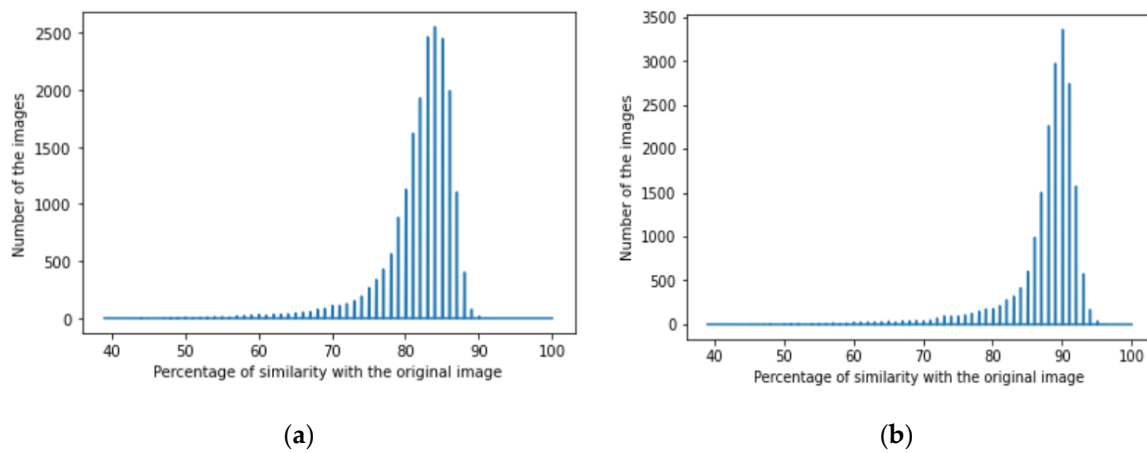


Figure 10. The histogram of similarity between the predicted denoised image and the original image in the case of speckle noise: (a) one autoencoder with separable convolutional neural networks; (b) dual autoencoder with separable convolutional neural networks.

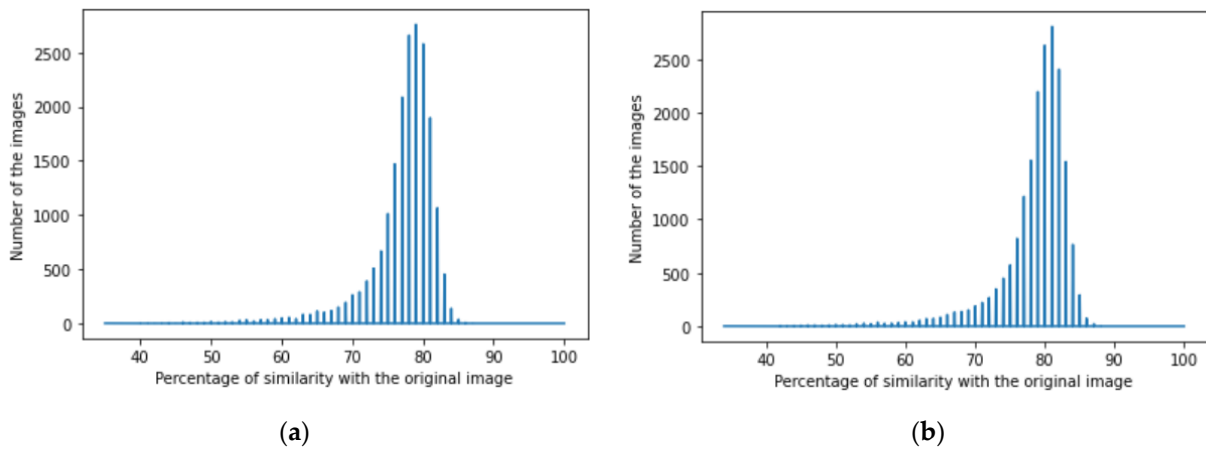


Figure 11. The histogram of similarity between the predicted denoised image and the original image in the case of impulse noise: (a) one autoencoder with convolutional neural networks; (b) dual autoencoder with convolutional neural networks.

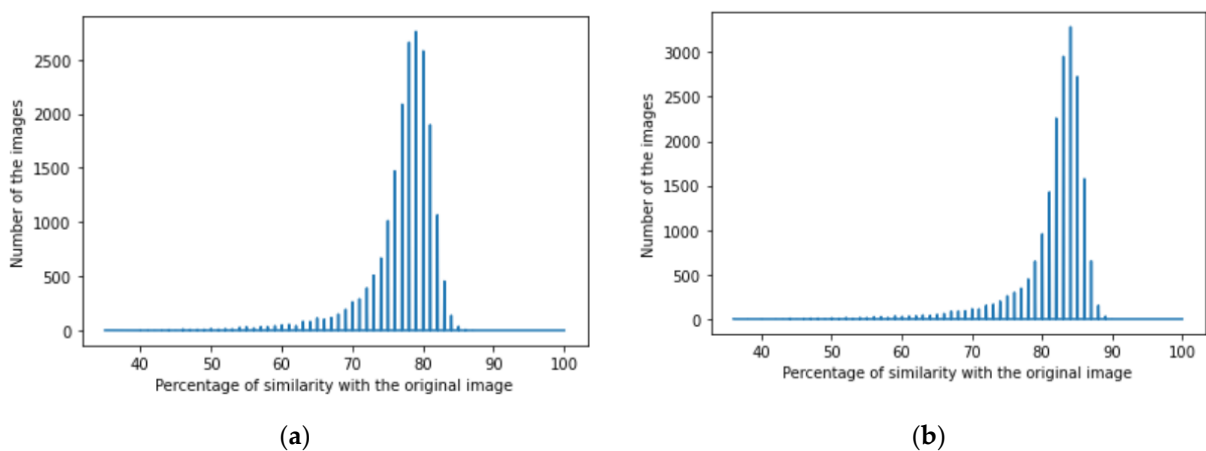


Figure 12. The histogram of similarity between the predicted denoised image and the original image in the case of impulse noise: (a) one autoencoder with separable convolutional neural networks; (b) dual autoencoder with separable convolutional neural networks.

The examples of the original image [8,68], the noisy image, and the images restored by one autoencoder (OA) and dual autoencoder (DA) with separable convolutional neural networks are shown in Figure 13a–d under Gaussian noise, Poisson noise, speckle noise, and impulse noise, respectively.



Figure 13. Original image [68–71], noisy image, and images restored by one and dual autoencoders with separable convolutional neural networks under (a) Gaussian noise, (b) Poisson noise, (c) speckle noise, and (d) impulse noise.

For one autoencoder and a dual autoencoder, the parameter numbers of the convolutions and separable convolutions, the total time of training 80,000 images during 40 epochs, the time of denoising of 19,538 images, the mode M of similarities, and the average PSNR after 40 epochs of training for the 19,538 testing images are noted in Tables 1–4 in the cases of Gaussian noise, Poisson noise, speckle noise, and impulse noise, respectively. In the tables, the statistical mode M (%) was the most frequent value of the similarity index percentage between the 19,538 denoised images and the original.

Table 1. Gaussian noise reduction.

Device of Image Denoising	Neural Network	Number of Parameters	Total Time of Training (s)	Time of Denoising(s)	M (%)	Average PSNR (dB)
Autoencoder	CNN	29,626	24,085	446	76	24.52
	SCNN	14,910	20,804	167	80	25.53
Dual Autoencoder	CNN	81,052	54,757	440	81	32.79
	SCNN	51,595	43,038	374	89	34.60

Table 2. Poisson noise reduction.

Device of Image Denoising	Neural Network	Number of Parameters	Total Time of Training (s)	Time of Denoising(s)	M (%)	Average PSNR (dB)
Autoencoder	CNN	29,626	22,001	60	71	27.38
	SCNN	14,910	19,412	58	71	27.77
Dual Autoencoder	CNN	81,052	56,324	55	76	30.59
	SCNN	51,595	41,207	49	77	31.11

Table 3. Speckle noise reduction.

Device of Image Denoising	Neural Network	Number of Parameters	Total Time of Training (s)	Time of Denoising(s)	M (%)	Average PSNR (dB)
Autoencoder	CNN	29,626	22,348	209	84	54.48
	SCNN	14,910	19,360	161	84	54.62
Dual Autoencoder	CNN	81,052	53,964	457	89	56.02
	SCNN	51,595	40,912	398	90	58.93

Table 4. Impulse noise reduction.

Device of Image Denoising	Neural Network	Number of Parameters	Total Time of Training (s)	Time of Denoising(s)	M (%)	Average PSNR (dB)
Autoencoder	CNN	29,626	24,052	184	78	54.93
	SCNN	14,910	20,293	173	79	55.07
Dual Autoencoder	CNN	81,052	55,708	480	85	58.34
	SCNN	51,595	42,007	460	85	58.03

As follows from the analysis of Figures 5–12 and Tables 1–4, the use of a dual autoencoder with separable convolutional neural networks for image denoising gave the following results:

- Image denoising for Gaussian noise (Gaussian blur) with a dual autoencoder achieved better results with both convolutional and separable convolutional neural networks where it raised the mode of the similarities of the testing dataset with the original by approximately 5% for convolution layers and 9% for separable convolution layers and the average PSNR by approximately 8 dB for convolution layers and 9 dB for separable convolution layers. A dual autoencoder with separable convolution layers achieved the higher accuracy with a value of 89% and with a high rate of performance, spending 0.019 s to denoise each image.
- Image denoising for Poisson noise with a dual autoencoder achieved better results for both the use of a convolutional neural network and separable convolutional neural network. Both kinds of neural networks achieved similar accuracy. However, the separable convolution layers enabled it faster with a lower number of parameters.
- Image denoising for speckle noise with a dual autoencoder raised the mode of the accuracies by approximately 6%, where a separable convolutional neural network and

convolutional neural network achieved the same results, but the performance of the separable neural network was faster. The average PSNR increased by approximately 2 dB for convolution layers and 4 dB for separable convolution layers. In the dual autoencoder, it took 0.02 s to denoise each image.

- Image denoising for impulse noise due to a dual autoencoder with both a convolutional neural network and separable convolutional neural network raised the M by 6% and the average PSNR by approximately 3 dB. Both kinds of neural networks achieved the same accuracy, but the separable convolutional neural network performed faster, where it took 0.023 s to denoise each image.

The research results in the following:

- A dual autoencoder ensures higher accuracy on increasing the total time of training and the time of denoising in comparison with one autoencoder. Thus, to reach a high accuracy of image denoising, we should use a more complicated device and spend a higher computational cost;
- In the mentioned situation, it would be better to have the approach that decreases computational cost without reducing the accuracy of restoring images. This way is a separable convolution in neural networks. In the cases of Poisson noise, speckle noise, and impulse noise, we kept the accuracy gained by the dual autoencoder and got a higher performance of this device by means of separable convolutional neural networks. In the case of Gaussian noise (Gaussian blur), we obtained both a higher accuracy and performance of the device, which is built on the basis of separable convolutional neural networks.

5. Discussion

Images frequently symbolize and say something of significance. They hold considerable information and are used for vast applications in security, scientific research, medicine, control, and other fields. In order for images to be beneficial data for these applications, they should be clear and clean from any noise. Therefore, our aim in this study was to solve this fundamental problem, which is image denoise and the restoration of the original images. In image processing and computer vision, we achieved a cancelation of Gaussian noise (Gaussian blur), Poisson noise, speckle noise, and impulse (salt and pepper) noise. To fulfill denoising, we proposed combining two autoencoders together to build a dual autoencoder, where the first autoencoder compresses the input image to approximately 23% of its original size and restores it while training to cancel the different noises, and the second autoencoder works as an enhancer that compresses the image to approximately 50% of its original size and restores it as the final output. Moreover, we used separable convolution layers to build the autoencoders for solving the denoising problem and compared them with the convolutional neural network to prove the ability of the separable convolutional neural network to maintain the same performance as the convolutional neural network and achieve better results while decreasing the number of learnable parameters and the processing time. From comparison with other types of neural networks, we highlight the following advantages of an autoencoder network:

- The ability of feature extraction from non-linear processes and performing non-linear transformations, for instance, image denoising.
- The ability of dimensionality reduction in the autoencoder network process of compressing the data and the feature of information retrievals in the autoencoder network process of decompressing the data.
- Reducing the training parameters decreases the training and performing time of the autoencoder's network and saves more memory on the user device.

The study's results show how the use of separable convolution layers in building autoencoders in the field of image processing and image denoising extends the opportunities for better quality performance, presented with the high similarity between the denoised images and the original images, lower complexness, and fast processing. At the same

time, a dual autoencoder with separable convolutional neural networks has a relationship design created as a grid of nodes where each pixel in the layer is associated with all the surrounding pixels. This design delivers feature extraction capability, permitting the pixels in the image to be defined as spatial with the locally and equally likely to occur extracted features at any position in the noisy input image. With the proceeds in the advancement of deep learning and the utilization of autoencoders as a model for finding distinctive solutions for specialized issues in image processing and computer vision, in addition to the need to decrease the processing time and memory utilization in numerous gadgets and devices such as microprocessors, phones, or pads, the utilization of separable convolutional neural networks in dual autoencoders will be extraordinary. Indeed, a dual autoencoder increases the accuracy of restoring images; however, it makes the process of encoding and decoding take twice the time of ordinary autoencoders. The separable convolutional neural network decreases the sum of learnable parameters and processing time to adjust the computation cost while achieving high quality and stability in a dual autoencoder.

6. Conclusions

In this study we used dual autoencoders for image denoising, which enhance the quality of the denoise where the first autoencoder works as a main image denoiser and the second completes the denoising process and enhances the quality of the denoised images. The use of a separable convolutional neural network in the field of building autoencoders for image processing and image denoising gives a higher performance of devices in comparison with a convolutional neural network. Moreover, for images affected by Poisson noise, speckle noise, and impulse noise, separable convolutional neural networks give the same accuracy of denoising as ordinary convolutional neural networks and better results in the case of Gaussian noise. With the continuous development of deep learning and the use of autoencoders as a tool for finding different solutions for technical problems in image processing and computer vision, and with the need to reduce processing time and memory usage in different devices such as processors, telephones, or pads, the use of a separable convolutional neural network will be signifiabile, especially with dual autoencoders, since a dual autoencoder duplicates the process of encoding and decoding. Moreover, separable convolutional neural networks decrease the amount of learnable parameters and processing time to balance the computation cost while achieving high quality results.

Author Contributions: Conceptualization, E.S.; methodology, E.S. and A.A.; software, A.A.; validation, E.S. and A.A.; formal analysis, E.S. and A.A.; investigation, E.S. and A.A.; resources, A.A.; data curation, A.A.; writing—original draft preparation, E.S. and A.A.; writing—review and editing, E.S. and A.A.; visualization, A.A.; supervision, E.S.; project administration, E.S.; funding acquisition, E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in publicly accessible repositories those do not issue DOIs. This data can be found here: IMDB-WIKI dataset is available at <https://data.vision.ee.ethz.ch/cv1/rrothe/imdb-wiki/> (accessed on 8 July 2022), FER-2013 dataset is available at <https://www.kaggle.com/msambare/fer2013> (accessed on 8 July 2022), KDEF dataset is available at <https://www.kdef.se/download-2/index.html> (accessed on 8 July 2022). Data available on request due to restrictions on use only for non-commercial research and educational purposes. This data can be found here: AffectNet dataset is available at <http://mohammadmahoor.com/affectnet/> (accessed on 8 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Roy, A.; Maity, P. A Comparative Analysis of Various Filters to Denoise Medical X-Ray Images. In Proceedings of the 2020 4th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), Kolkata, India, 2–4 October 2020. [\[CrossRef\]](#)
2. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Virtual Conference, Montreal, BC, Canada, 11–17 October 2021. [\[CrossRef\]](#)
3. Liu, X.; Suganuma, M.; Sun, Z.; Okatani, T. Dual Residual Networks Leveraging the Potential of Paired Operations for Image Restoration. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
4. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H.; Shao, L. Multi-Stage Progressive Image Restoration. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021. [\[CrossRef\]](#)
5. Solovyeva, E. Cellular Neural Network as a Non-linear Filter of Impulse Noise. In Proceedings of the 2017 20th Conference of Open Innovations Association FRUCT (FRUCT20), St. Petersburg, Russia, 3–7 April 2017; pp. 420–426. [\[CrossRef\]](#)
6. Solovyeva, E.B.; Degtyarev, S.A. Synthesis of Neural Pulse Interference Filters for Image Restoration. *Radioelectron. Commun. Syst.* **2008**, *51*, 661–668. [\[CrossRef\]](#)
7. Solovyeva, E. Types of Recurrent Neural Networks for Non-linear Dynamic System Modelling. In Proceedings of 2017 IEEE International Conference on Soft Computing and Measurements (SCM2017), St. Petersburg, Russia, 24–26 May 2017; pp. 252–255. [\[CrossRef\]](#)
8. Kim, J.S.; Chang, D.S.; Choi, Y.S. Enhancement of Multi-Target Tracking Performance via Image Restoration and Face Embedding in Dynamic Environments. *Appl. Sci.* **2021**, *11*, 649. [\[CrossRef\]](#)
9. Yang, L.; Fan, J.; Huo, B.; Li, E.; Liu, Y. Image Denoising of Seam Images With Deep Learning for Laser Vision Seam Tracking. *IEEE Sens. J.* **2022**, *22*, 6098–6107. [\[CrossRef\]](#)
10. Leng, C.; Zhang, H.; Cai, G.; Chen, Z.; Basu, A. Total Variation Constrained Non-Negative Matrix Factorization for Medical Image Registration. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1025–1037. [\[CrossRef\]](#)
11. Jia, X.; Thorley, A.; Chen, W.; Qiu, H.; Shen, L.; Styles, I.B.; Duan, J. Learning a Model-Driven Variational Network for Deformable Image Registration. *IEEE Trans. Med. Imaging* **2022**, *41*, 199–212. [\[CrossRef\]](#)
12. Kollem, S.; Reddy, K.R.L.; Rao, D.S. A Review of Image Denoising and Segmentation Methods Based on Medical Images. *Int. J. Mach. Learn. Comput.* **2019**, *9*, 288–295. [\[CrossRef\]](#)
13. Tian, C.; Chen, Y. Image Segmentation and Denoising Algorithm Based on Partial Differential Equations. *IEEE Sens. J.* **2020**, *20*, 11935–11942. [\[CrossRef\]](#)
14. Garg, A.; Khandelwal, V. Combination of Spatial Domain Filters for Speckle Noise Reduction in Ultrasound Medical Images. *Adv. Electr. Electron. Eng.* **2018**, *15*, 857–865. [\[CrossRef\]](#)
15. Arabi, H.; Zaidi, H. Improvement of image quality in PET using post-reconstruction hybrid spatial-frequency domain filtering. *Phys. Med. Biol.* **2018**, *63*, 215010. [\[CrossRef\]](#)
16. Huang, Z.; Zhang, Y.; Li, Q.; Zhang, T.; Sang, N. Spatially adaptive denoising for X-ray cardiovascular angiogram images. *Biomed. Signal Processing Control* **2018**, *40*, 131–139. [\[CrossRef\]](#)
17. Suresh, S.; Lal, S.; Chen, C.; Celik, T. Multispectral Satellite Image Denoising via Adaptive Cuckoo Search-Based Wiener Filter. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4334–4345. [\[CrossRef\]](#)
18. Hasan, M.; El-Sakka, M.R. Improved BM3D image denoising using SSIM-optimized Wiener filter. *EURASIP J. Image Video Processing* **2018**, *25*, 1–12. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Ishikawa, A.; Tajima, H.; Fukushima, N. Halide implementation of weighted median filter. In Proceedings of the International Workshop on Advanced Imaging Technology (IWAIT) 2020, Yogyakarta, Indonesia, 5–7 January 2020. [\[CrossRef\]](#)
20. Mafi, M.; Rajaei, H.; Cabrerizo, M.; Adjouadi, M. A Robust Edge Detection Approach in the Presence of High Impulse Noise Intensity Through Switching Adaptive Median and Fixed Weighted Mean Filtering. *IEEE Trans. Image Processing* **2018**, *27*, 5475–5490. [\[CrossRef\]](#)
21. Iqbal, N.; Ali, S.; Khan, I.; Lee, B. Adaptive Edge Preserving Weighted Mean Filter for Removing Random-Valued Impulse Noise. *Symmetry* **2019**, *11*, 395. [\[CrossRef\]](#)
22. Ben Said, A.; Hadjidj, R.; Fofou, S. Total Variation for Image Denoising Based on a Novel Smart Edge Detector: An Application to Medical Images. *J. Math. Imaging Vis.* **2018**, *61*, 106–121. [\[CrossRef\]](#)
23. Hanh, D.N.H.; Thanh, L.T.; Hien, N.N.; Prasath, S. Adaptive total variation L1 regularization for salt and pepper image denoising. *Optik* **2020**, *208*, 163677. [\[CrossRef\]](#)
24. Zhang, H.; Liu, L.; He, W.; Zhang, L. Hyperspectral Image Denoising With Total Variation Regularization and Nonlocal Low-Rank Tensor Decomposition. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3071–3084. [\[CrossRef\]](#)
25. Bal, A.; Banerjee, M.; Chaki, R.; Sharma, P. An efficient method for PET image denoising by combining multi-scale transform and non-local means. *Multimed. Tools Appl.* **2020**, *79*, 29087–29120. [\[CrossRef\]](#)
26. Wang, G.; Liu, Y.; Xiong, W.; Li, Y. An improved non-local means filter for color image denoising. *Optik* **2018**, *173*, 157–173. [\[CrossRef\]](#)

27. Heo, Y.-C.; Kim, K.; Lee, Y. Image Denoising Using Non-Local Means (NLM) Approach in Magnetic Resonance (MR) Imaging: A Systematic Review. *Appl. Sci.* **2020**, *10*, 7028. [[CrossRef](#)]
28. Ravishankar, S.; Ye, J.C.; Fessler, J.A. Image Reconstruction: From Sparsity to Data-Adaptive Methods and Machine Learning. *Proc. IEEE* **2020**, *108*, 86–109. [[CrossRef](#)] [[PubMed](#)]
29. Madathil, B.; George, S.N. DCT based weighted adaptive multi-linear data completion and denoising. *Neurocomputing* **2018**, *318*, 120–136. [[CrossRef](#)]
30. Kuttan, D.B.; Kaur, S.; Goyal, B.; Dogra, A. Image Denoising: Pre-processing for enhanced subsequent CAD Analysis. In Proceedings of the 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 7–9 October 2021. [[CrossRef](#)]
31. Sawant, A.; Kulkarni, S. Hybrid Filtering Techniques For Medical Image Denoising. In Proceedings of the International Conference on Business Management, Innovation, and Sustainability (ICBMIS-2020), Dubai, United Arab Emirates, 13–14 April 2020; pp. 1–10. [[CrossRef](#)]
32. Zhou, L.; Schaefferkoetter, J.D.; Tham, I.W.K.; Huang, G.; Yan, J. Supervised learning with cycleGAN for low-dose FDG PET image denoising. *Med. Image Anal.* **2020**, *65*, 101770. [[CrossRef](#)] [[PubMed](#)]
33. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.-W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. [[CrossRef](#)]
34. Tian, C.; Xu, Y.; Fei, L.; Yan, K. Deep Learning for Image Denoising: A Survey. *Genet. Evol. Comput.* **2019**, *834*, 563–572. [[CrossRef](#)]
35. Pang, T.; Zheng, H.; Quan, Y.; Ji, H. Recorrupted-to-Recorrupted: Unsupervised Deep Learning for Image Denoising. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021. [[CrossRef](#)]
36. Ignatov, A.; Byeoung-Su, K.; Timofte, R.; Pouget, A.; Song, F.; Li, C.; Chen, F. Fast Camera Image Denoising on Mobile GPUs with Deep Learning, Mobile AI 2021 Challenge: Report. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021. [[CrossRef](#)]
37. Cui, J.; Gong, K.; Guo, N.; Wu, C.; Meng, X.; Kim, K.; Li, Q. PET image denoising using unsupervised deep learning. *Eur. J. Nucl. Med. Mol. Imaging* **2019**, *46*, 2780–2789. [[CrossRef](#)]
38. Hong, I.; Hwang, Y.; Kim, D. Efficient deep learning of image denoising using patch complexity local divide and deep conquer. *Pattern Recognit.* **2019**, *96*, 106945. [[CrossRef](#)]
39. Islam, M.T.; Rahman, S.M.; Ahmad, M.O.; Swamy, M.N.S. Mixed Gaussian-impulse noise reduction from images using convolutional neural network. *Signal Processing Image Commun.* **2018**, *68*, 26–41. [[CrossRef](#)]
40. Ilesanmi, A.E.; Ilesanmi, T.O. Methods for image denoising using convolutional neural network: A review. *Complex Intell. Syst.* **2021**, *7*, 2179–2198. [[CrossRef](#)]
41. Paul, E.; Sabeenian, R.S. Modified convolutional neural network with pseudo-CNN for removing nonlinear noise in digital images. *Displays* **2022**, *74*, 102258. [[CrossRef](#)]
42. Munir, N.; Park, J.; Kim, H.-J.; Song, S.-J.; Kang, S.-S. Performance enhancement of convolutional neural network for ultrasonic flaw classification by adopting autoencoder. *NDT E Int.* **2020**, *111*, 102218. [[CrossRef](#)]
43. Singha, S.; Sk. Imran Hossain, M.A.H.; Murase, K. A Robust System for Noisy Image Classification Combining Denoising Autoencoder and Convolutional Neural Network. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 224–235. [[CrossRef](#)]
44. Paul, A.; Kundu, A.; Chaki, N.; Dutta, D.; Jha, C.S. Wavelet enabled convolutional autoencoder based deep neural network for hyperspectral image denoising. *Multimed. Tools Appl.* **2021**, *81*, 2529–2555. [[CrossRef](#)]
45. Saleh Ahmed, A.; El-Behaidy, W.H.; Youssif, A.A.A. Medical image denoising system based on stacked convolutional autoencoder for enhancing 2-dimensional gel electrophoresis noise reduction. *Biomed. Signal Processing Control* **2021**, *69*, 102842. [[CrossRef](#)]
46. Kang, G.; Gao, S.; Yu, L.; Zhang, D. Deep Architecture for High-Speed Railway Insulator Surface Defect Detection: Denoising Autoencoder With Multitask Learning. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 2679–2690. [[CrossRef](#)]
47. Awad, A. Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise. *Eng. Sci. Technol.* **2019**, *22*, 746–753. [[CrossRef](#)]
48. Abubakar, A.; Zhao, X.; Li, S.; Takruri, M.; Bastaki, E.; Bermak, A. A Block-Matching and 3-D Filtering Algorithm for Gaussian Noise in DoFP Polarization Images. *IEEE Sens. J.* **2018**, *18*, 7429–7435. [[CrossRef](#)]
49. Kumain, S.C.; Singh, M.; Singh, N.; Kumar, K. An efficient Gaussian Noise Reduction Technique For Noisy Images using optimized filter approach. In Proceedings of the 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018. [[CrossRef](#)]
50. Khan, K.B.; Shahid, M.; Ullah, H.; Rehman, E.; Khan, M.M. Adaptive trimmed mean autoregressive model for reduction of poisson noise in scintigraphic images. *IIUM Eng. J.* **2018**, *19*, 68–79. [[CrossRef](#)]
51. Kumar, P.G.; Sahay, R.R. Low Rank Poisson Denoising (LRPD): A Low Rank Approach Using Split Bregman Algorithm for Poisson Noise Removal From Images. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–20 June 2019. [[CrossRef](#)]
52. Kumar, M.; Mishra, S.K. Jaya based functional link multilayer perceptron adaptive filter for Poisson noise suppression from X-ray images. *Multimed. Tools Appl.* **2018**, *77*, 24405–24425. [[CrossRef](#)]
53. Duarte-Salazar, C.A.; Castro-Ospina, A.E.; Becerra, M.A.; Delgado-Trejos, E. Speckle Noise Reduction in Ultrasound Images for Improving the Metrological Evaluation of Biomedical Applications: An Overview. *IEEE Access* **2020**, *8*, 15983–15999. [[CrossRef](#)]

54. Kumar Pal, S.; Bhardwaj, A.; Shukla, A. A Review on Despeckling Filters in Ultrasound Images for Speckle Noise Reduction. In Proceedings of the 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 4–5 March 2021. [CrossRef]
55. Chen, Z.; Zeng, Z.; Shen, H.; Zheng, X.; Dai, P.; Ouyang, P. DN-GAN: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images. *Biomed. Signal Processing Control* **2020**, *55*, 101632. [CrossRef]
56. Garg, B. Restoration of highly salt-and-pepper-noise-corrupted images using novel adaptive trimmed median filter. *Signal Image Video Processing* **2020**, *14*, 1555–1563. [CrossRef]
57. Solovyeva, E. A Split Signal Polynomial as a Model of an Impulse Noise Filter for Speech Signal Recovery. *J. Phys. Conf. Ser. (JPCS)* **2017**, *803*, 012156. [CrossRef]
58. Solovyeva, E. Operator Approach to Nonlinear Compensator Synthesis for Communication Systems. In Proceedings of the 2016 International Siberian Conference on Control and Communications (SIBCON), Moscow, Russia, 12–14 May 2016; pp. 1–5. [CrossRef]
59. Garg, B.; Arya, K.V. Four stage median-average filter for healing high density salt and pepper noise corrupted images. *Multimed. Tools Appl.* **2020**, *79*, 32305–32329. [CrossRef]
60. Varatharajan, R.; Vasanth, K.; Gunasekaran, M.; Priyan, M.; Gao, X.Z. An adaptive decision based kriging interpolation algorithm for the removal of high density salt and pepper noise in images. *Comput. Electr. Eng.* **2018**, *70*, 447–461. [CrossRef]
61. Choi, H.; Kim, M.; Lee, G.; Kim, W. Unsupervised learning approach for network intrusion detection system using autoencoders. *J. Supercomput.* **2019**, *75*, 5597–5621. [CrossRef]
62. Park, S.; Yu, S.; Kim, M.; Park, K.; Paik, J. Dual Autoencoder Network for Retinex-Based Low-Light Image Enhancement. *IEEE Access* **2018**, *6*, 22084–22093. [CrossRef]
63. Drozdov, A.; Rongali, S.; Chen, Y.-P.; O’Gorman, T.; Iyyer, M.; McCallum, A. Unsupervised Parsing with S-DIORA: Single Tree Encoding for Deep Inside-Outside Recursive Autoencoders. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual Conference, Punta Cana, Dominican Republic, 16–20 November 2020. [CrossRef]
64. Dong, G.; Liao, G.; Liu, H.; Kuang, G. A Review of the Autoencoder and Its Variants: A Comparative Perspective from Target Recognition in Synthetic-Aperture Radar Images. *IEEE Geosci. Remote Sens. Mag.* **2018**, *6*, 44–68. [CrossRef]
65. Lopez Pinaya, W.H.; Vieira, S.; Garcia-Dias, R.; Mechelli, A. Autoencoders. In *Machine Learning*; Mechelli, A., Vieira, S., Eds.; Elsevier Science: Amsterdam, The Netherland, 2019; pp. 193–208. [CrossRef]
66. Solovyeva, E.; Abdullah, A. Binary and Multiclass Text Classification by Means of Separable Convolutional Neural Network. *Inventions* **2021**, *6*, 70. [CrossRef]
67. Gao, H.; Yuan, H.; Wang, Z.; Ji, S. Pixel Transposed Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1218–1227. [CrossRef]
68. Rothe, R.; Timofle, R.; Gool, L.V. IMDB-WIKI Dataset. 2015. Available online: <https://data.vision.ee.ethz.ch/cv1/rrothe/imdb-wiki/> (accessed on 11 July 2022).
69. Sambare, M. FER-2013 Dataset. 2020. Available online: <https://www.kaggle.com/msambare/fer2013> (accessed on 11 July 2022).
70. Lundqvist, D.; Flykt, A.; Ohman, A. KDEF Dataset. 1998. Available online: <https://www.kdef.se/download-2/index.html> (accessed on 11 July 2022).
71. Mahoor, M.H. AffectNet Dataset. 2017. Available online: <http://mohammadmahoor.com/affectnet/> (accessed on 11 July 2022).
72. Shukla, R.K.; Agarwal, A.; Malviya, A.K. An Introduction of Face Recognition and Face Detection for Blurred and Noisy Images. *Int. J. Sci. Res. Comput. Sci. Eng.* **2018**, *6*, 39–43. [CrossRef]
73. Chen, S.; Zhang, Y.; Li, Y.; Chen, Z.; Wang, Z. Spherical Structural Similarity Index for Objective Omnidirectional Video Quality Assessment. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018. [CrossRef]
74. Luo, J.; Lei, W.; Hou, F.; Wang, C.; Ren, Q.; Zhang, S.; Luo, S.; Wang, Y.; Xu, L. GPR B-Scan Image Denoising via Multi-Scale Convolutional Autoencoder with Data Augmentation. *Electronics* **2021**, *10*, 1269. [CrossRef]