Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Image Quality Enhancement using Deep learning-based Convolution Residual Networks Technique

Bhavana Sharma[1*], Hukum Singh[1], Mehak Khurana[2]
[1*]The NorthCap University, Gurugram, Haryana,India
[2]College of Information Technology, Ahlia University, Manama, Bahrain
***Corresponding Author:** Bhavana Sharma
*Email: drbhavnasharma2023@gmail.com

ARTICLE INFO

Received: 17 Dec 2024

Revised: 14 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

A new approach for cleaning encrypted images is developed using Deep Convolutional Residual Network (Deep ConvResNet) as the proposed method. The aim of this research is to protect encrypted images from noise attacks by utilizing ResNet denoising capabilities. It has been proven that ResNets are successful at cleaning up noise while maintaining the important picture characteristics. This research employs multiple datasets for training and performs a detailed comparative study using peak signal to noise ratio and structure similarity index as well as noise and occlusion and blur attack metrics. Results of the simulation show that the suggested cryptosystem is resistant to the familiar attacks. Filtering based denoising techniques and CNNs are worse than ResNets because ResNets have better efficiency and resilience to occlusion and noise and blur attacks. The graphs of training loss vs. epoch show the convergence pattern of the model during training. Considering its potential use, this methodology is applicable in secure image transmission in different domains such as healthcare and multimedia transmission.

KEYWORDS: Convolutional Neural Network, Image Denoising, Deep Learning, Epoch, Residual Network, Computer Vision.

1. INTRODUCTION

Today's digital world has increased the demand for securing image transmission and protection against noise attacks. Image encryption plays a vital role in protecting data during transmission. Techniques like Double Random Phase Encoding (DRPE) [Refregier & Javidi, 1995] and Fractional Fourier Transform (FrFT) [Unnikrishnan et al., 2000] offer high security but remain vulnerable to noise attacks, which degrade image quality during transmission and decryption.

Although existing denoising techniques such as Convolutional Neural Networks (CNNs) [Zhang et al., 2017] and Multilayer Perceptrons (MLPs) [Burger et al., 2012] have made remarkable progress, they still face challenges in handling noise effectively. Recent advances in image denoising through deep learning-based models such as FFDNet and DnCNN [Zhang et al., 2018] have shown significant promise in noise reduction. However, these methods often fall short in effectively handling encrypted images due to limitations in preserving fine image details under diverse noise scenarios [Dong et al., 2019; Li et al., 2023].

The core focus of this paper is on addressing the unique complexities of denoising encrypted images. Unlike conventional methods that struggle with preserving image details, our model leverages deeper network architectures and advanced deep learning techniques, particularly the Deep Convolutional Residual Network (Deep ConvResNet), to achieve superior denoising performance. ResNet-based methods [Ren et al., 2019; Zhang et al., 2021] not only efficiently denoise images but also defend against noise, occlusion, and blur attacks.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

This makes ResNet architecture particularly suitable for denoising encrypted images in secure multimedia applications, including healthcare and critical infrastructure domains [Chen et al., 2019]. Furthermore, experimental results are presented comparing our model with other leading noise filtering techniques, highlighting its superior performance in terms of image quality and noise removal efficiency.

This paper's structure is as follows: after defining key concepts in image encryption and denoising, we discuss the proposed Deep ConvResNet model and its architecture. The experimental setup, including datasets, hardware, and evaluation metrics, is then presented. Finally, we conclude with a discussion of the results and potential future work.

Different deep learning based image denoising techniques are compared in Table 1. The improvements made by the proposed approach over existing methods include dealing with the limitations like computational complexity, image quality compromise as well as adaptability to various noise patterns. Unlike earlier studies, this ensures the robustness of denoising while keeping the efficiency and real time ability.

Table 1: Comparison of various image denoising models based on deep learning

Study	Method Used	Description	Drawbacks	Comparison with this Paper
Jing et al., 2022	LDCT	Denoising low dose Computed Tomography	Compromise in image quality thus images are challenging	This study has better performance in noise, blur and occlusion attacks.
Tian et al., 2020	TDL-CDL	combines CDL and CNNs and uses a compound loss function to strike a compromise between detail retention and noise reduction.	May struggle with high noise levels, complex parameter tuning	This study uses ResNets based residual learning, showing better performance in noise, blur and occlusion attacks.
Xu et al., 2023	DUMRN	Learns deep prior information from multi-resolution features by eliminating noise in the deep feature space using its feature-based denoising module with a multi-scale regularise block.	This is computational intensive thus might not be suitable for real time applications	This study just not produce comparable result but also denoise with a focus on efficiency and has real-time application potential
Singh et al., 2023	DL-based autoencoder	Targets speckle noise in ultrasonic images, outperforms BM3D	Limited to specific noise type and thus is less versatile	This new model can handle more types of noise(including blur and occlusion) effectively.
Huang et al., 2023	MD3 to denoise remote sensing images (RSIs)	Uses a learnt dictionary and a rearranged self-similar data matrix	It does not only have computational overhead but also have scalability issue	This study uses efficient computation and make it more scalable
X. Wei et al., 2023	DIBD	It can handle complex real-world noise scenarios using dual learning algorithm	Complex framework thus requires extensive data for learning joint distribution	This study is more adaptable and simpler also providing robust denoising across various image types without extensive data requirenment

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

K.Zhang et al., 2017	Image Restoration CNN	CNN-based denoising for image restoration	Less effective in handling complex noise patterns, requires improvements for efficient reconstruction	This study addresses these issues. Showing superior noise patterns and enhancing image quality

RELATED TERMS

1.1. Convolutional Neural Network

CNNs establish themselves as basic deep learning procedures that perform extensively in tasks related to computer vision, such as segmenting and detecting objects and classifying images. A CNN architecture contains Three primary layers comprise convolutional, pooling, and fully linked layers. The different layers function as a unified system to process input data while creating output results. Q. Zhang et al. (2023) developed the convolutional layer which applies filters to images to detect edge patterns and textures with the subsequent output of feature maps that show these detected patterns. The pooling layers decrease feature map dimensions through maximum or average value selection from localized areas which reduces model complexity and improves its ability to handle input variations. Yuan et al. (2019) describes fully connected layers which transform the flattened feature map into output classes through SoftMax function-based probability distribution generation.

1.2. Residual Network

ResNet serves as one of the most popular architectures within computer vision applications specifically designed for image denoising. The residual blocks of ResNet contain multiple convolutional layers which are connected through skip connections. The direct transmission of information through skip connections in the network helps resolve the issue of the diminishing gradient according to Ren et al. (2019). A residual block of ResNet appears in Figure 1 with the addition operator receiving the layer input x through a solid line that represents the shortcut connection. The training of deep networks for image denoising tasks becomes possible through this method while simultaneously resolving the gradient disappearance issue (J. Zhang et al., 2021). The layers inside ResNet contain two separate pathways which include a standard feedforward path that applies convolutional and activation layers and a shortcut path that jumps over selected convolutional layers (Maharjan et al., 2019). The residual network structure is illustrated in Equation 1 which shows the input x as well as the output f(x).

$$\boldsymbol{f(x) = g(x) + x} \qquad \textbf{(1)}$$

Residual blocks enable input data to move more quickly between network layers through their connection mechanism. The network uses bottleneck layers to decrease computational complexity and parameter count. Figure 2 shows two different types of convolutional residual blocks. The basic block as shown in Figure 2(a) involves two 3×3 convolutions, while the bottleneck block in Figure 2(b) integrates an extra 1×1 convolution for dimension reduction and restoration which optimizes the efficiency of the subsequent 3×3 convolution. The initial step in this process involves the application of 1×1 convolutional layer, which minimize the channel count in the data. This is subsequently followed by a 3×3 convolutional layer to further refine the information. Lastly, return the number of channels to its initial level in the data, another 1×1 convolutional layer is applied. The ResNet block without a 1×1 convolution is employed in less deep ResNet architectures or situations where computational efficiency is a priority over learning intricate features. This reduction in parameters results in a more manageable training process. However, it might possess slightly less representation capacity when compared to the bottleneck block. The additional addition is done to optimize the performance of deeper ResNet models. Which allows effective training of large-scale networks with enhanced efficiency. ResNet has achieved significant success in image classification, surpassing previous benchmarks (F. Wei et al., 2023). It uses of residual blocks and skip connections enabling effective training which makes it a valuable tool in deep

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

learning.

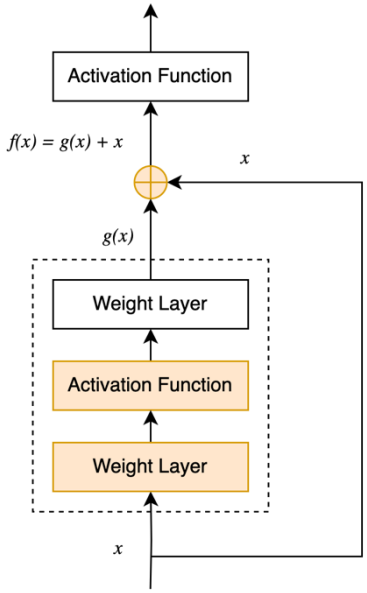


Figure 1. Residual network: block diagram

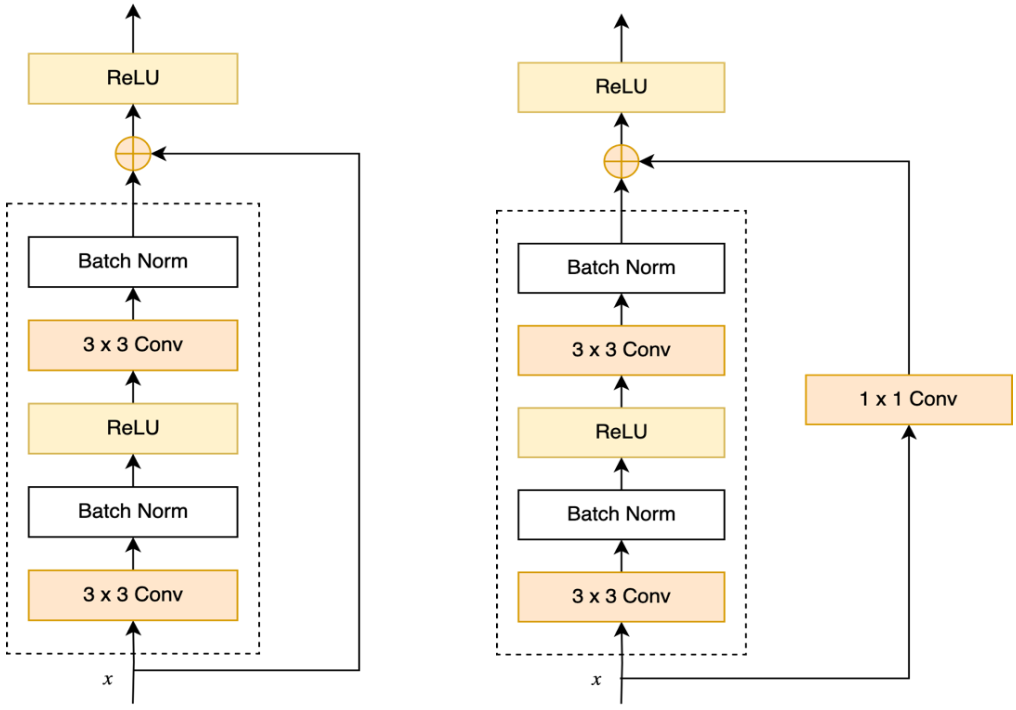


Figure 2. ResNet block, both with and without a 1x1 convolution, which modifies the input to take on the desired shape before adding operation.

1.3. Fractional Fourier Transform

Fractional Fourier Transform (FrFT) is a mathematical technique that generalizes the standard Fourier Transform, enabling rotation of functions in the time-frequency domain. The signal processing field uses this technique frequently (Khurana & Singh, 2020). The FrFT operates through a kernel structure that contains the complex exponential function elevated to fractional powers for forward and reverse transform operations.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Researchers employ FrFT because it applies to optical imaging techniques whereas it enhances resolutions and lowers noise levels. A specific FrFT operation on a function appears as follows:

$$f(x)(u) = \int_{-\infty}^{\infty} K_\alpha(x,u)\, f(x)dx \quad (2)$$

Where K_α is given as,

$$K_\alpha(x,u) = \frac{exp\left\{i\left[\frac{\pi}{4}sgn(\sin(\Phi)) - \frac{\Phi}{2}\right]\right\}}{\sqrt{|sin(\Phi)|}} exp\{i\pi((u^2 + x^2)cot(\Phi) - 2uxcsc(\Phi))\} \quad (3)$$

The signum function sgn appears in equation (3) together with the angle $\Phi=\pi\alpha/2$ that represents the transform order α.

1.4. Pixel Scrambling

Image encryption is done through a method called Pixel Scrambling in order to protect the privacy of images. It consists of splitting the image into blocks and scrambling the pixel positions inside each block by pseudo random permutations. Then, the final scramble image is formed by concatenating the scrambled blocks.

2. PROPOSED MODEL

2.1. The Deep Conv ResNet Denoising Model

Image denoising is performed with the proposed model using deep learning algorithm based on residual networks (ResNet). This architecture has the output of each convolutional layer linked to every nth deconvolutional layer which makes it an efficient learning and feature extraction. The process is divided into three steps: encryption, decryption and denoising of the unencrypted image. As represented in Figure 3.

The model is trained on several datasets such as Waterloo's Pristine image, McMaster's, and MCBSD68 datasets. The data is divided in an 80:20 ratio between training and testing sets for preprocessing. To augment the effectiveness of the suggested deep neural network, the methods of dataset tagging, normalisation and standardisation are used. We compare various image restoration architectures such as autoencoders, CNN based image restoration (Raj et al., 2020), Multilayer Perceptrons (MLP) (Burger et al., 2012), Block Matching 3D Filtering (BM3D) (Burger et al., 2012) and ConvLSTM (Piriyatharawet et al., 2018). These techniques are employed to simulate and assess the suggested model's performance.

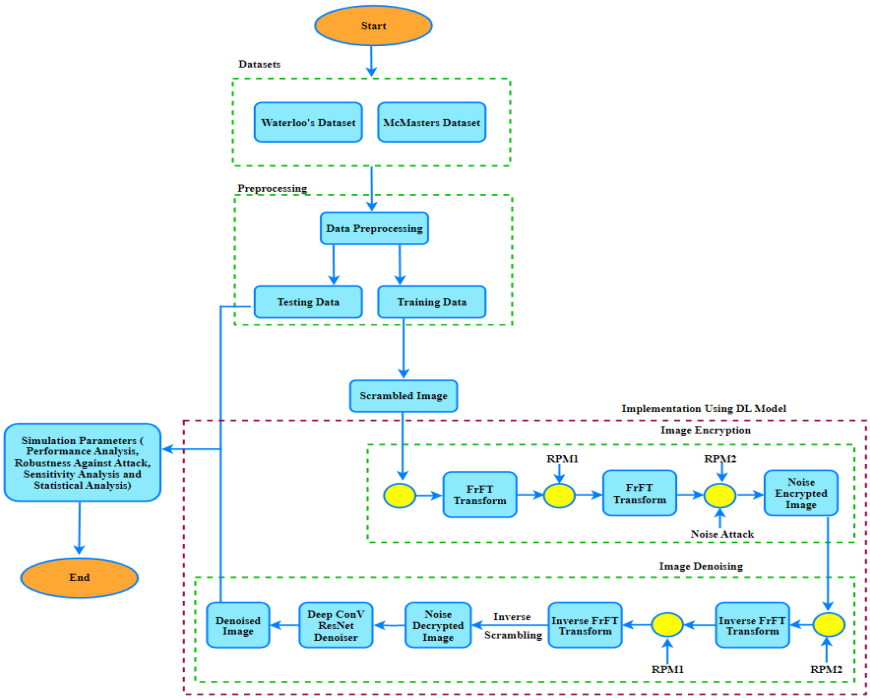


Figure 3. Building Deep Learning Model

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2.2. Optical Image Encryption Based on Fractional Fourier Transform (FrFT)

The role of image encryption is very important in data security (Arora & Khurana, 2020). A new encryption method is proposed in the current study, in which the encryption process starts with pixel scrambling using a private key for security enhancement. Then, a Fractional Fourier Transform (FrFT) is applied with fractional orders α and β. The following subsection gives details on the encryption and decryption steps.

2.2.1. Encryption Process

Below we discuss the process of encryption, refer to Figure 4(a) for the Schematic of the proposed encryption model.

1.The original picture $I(x, y)$ that needs to be encrypted is first scrambled using the private key generated from the pixel scrambling (PK) to obtain $I_1(u, v)$ as shown in equation (4).

$$\boldsymbol{I_1(u,v)} = [\boldsymbol{I(x,y)} \times \boldsymbol{PK}] \quad \textbf{(4)}$$

2. The scrambled image $I_2(x, y)$ is then convoluted by first random phase mask RPM_1 in fractional Fourier domain with rotational angle α as shown in equation (5) where $RPM = exp\,(2\pi i * v(x, y))$ and $v(x, y)$ is any random matrix with the same dimensions as the input image

$$\boldsymbol{H(u'v')} = \boldsymbol{FrFT}^{\boldsymbol{\alpha,\beta}}[\boldsymbol{I_1(u,v)} \times \boldsymbol{RPM_1}] \quad \textbf{(5)}$$

3. The obtain the final encrypted image $E(x, y)$, the intermediate result $H(u'v')$ is convoluted with another RPM, RPM_2 and propagated through inverse fractional Fourier transform domain with rotational angle β as shown in equation (6) in step 3.

$$\boldsymbol{E(x,y)} = \boldsymbol{FrFT}^{-\alpha,-\beta}\ [\boldsymbol{H(u'v')} \times \boldsymbol{RPM_2}\ ] \quad \textbf{(6)}$$

4. The encrypted image is artificially targeted with noise during the encryption phase to mimic an actual noise attack situation on the encryption device, as presented in Figure 4(a) and shown in equation (7).

$$\boldsymbol{E'(x,y)} = \boldsymbol{E(x,y)} + \boldsymbol{Noise} \quad \textbf{(7)}$$

The quality of the encrypted image is distorted in real-world situations when noises like AWGN, salt-and-pepper, speckle, and Poisson noise are introduced during transmission (X. Wei et al., 2023).

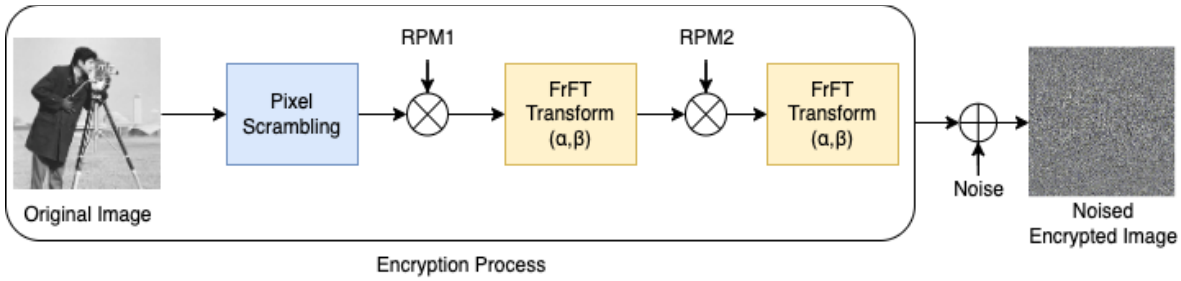


Figure 4(a). Schematic of the proposed encryption model

2.2.2. Decryption Process

The decryption process is the inverse of the encryption process (Anshula & Singh, 2021; Ding et al., 2021) and is presented in Figure 4(b).

1.During the propagation process, the encrypted image, which is distorted by noise and referred to as $E'(x, y)$ is convolved employing the conjugate of the second random phase mask (RPM_2^*) while utilizing Fractional Fourier Transform and resultant image is obtained is $H(u'v')$ as shown in equation (8).

$$\boldsymbol{H(u'v')} = \boldsymbol{FrFT}^{\boldsymbol{\alpha,\beta}}[\boldsymbol{E'(x,y)} \times \mathbf{RPM}_2^*] \quad \textbf{(8)}$$

2. Then the resultant image $H(u'v')$ is convoluted with the conjugate of RPM_1^* in inverse Fractional Fourier Transform domain with rotational angle β to obtain $H(u, v)$, as shown below in equation (9).

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$H(u,v) = FrFT^{-\alpha,-\beta}[H(u'v') \times \mathbf{RPM}_1^*] \quad \textbf{(9)}$$

3. Finally, in order attain the decrypted image $I(x,y)$, one must apply the inverse of the (PK), i.e Private key of scrambling algorithm to the resultant image $H(u,v)$, as shown in equation (10).

$$I(x,y) = [H(u,v) \times IPK] \quad \textbf{(10)}$$

Finally, the decrypted image denoted by $I(x,y)$ is obtained. After decryption, a residual model is utilized for denoising to eliminate any introduced noise and boost the image quality. The steps of denoising are presented in next section and denoiser illustrated in Figure 4(b).

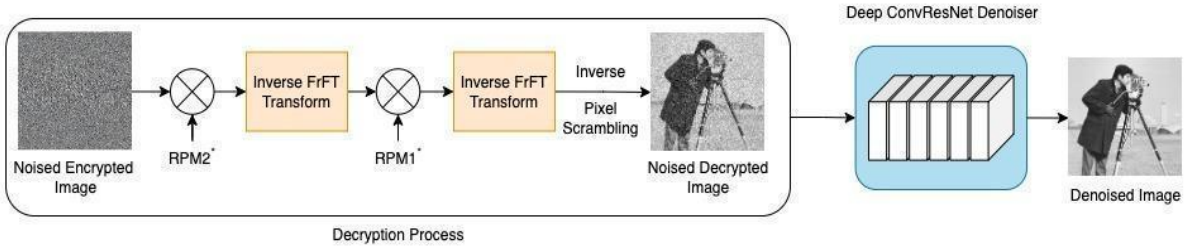


Figure 4(b). Schematic of the proposed decryption model

2.3. Decrypted Image Denoiser

The proposed deep ConvResNet denoiser utilizes a deep residual network (ResNet) architecture for denoising decrypted images. The ResNet denoiser consists of nine layers, including convolutional and deconvolutional layers, as shown in Figure 5(a).

3.3.1 Architecture of the Model

There are 9 convolutional layers in the architecture which are divided into four distinct blocks. The initial block of the architecture consisted of two layers, particularly a Conv2D layer that operates in two dimensions, followed by an activation function that implements the Leaky Rectified Linear Unit (LeakyReLU) approach. LeakyReLU is chosen as the activation functionfor all convolutional layers to mitigate the Dying ReLU problem. LeakyReLU lower risk of overfitting, computational efficiency and consistent performance across Various scenarios make it a better option when compared with PreLU.In addition to leverage Empirical performance Improvements, improving convergence rates and maintaining a balance between simplicity and performance efficiency offered by LeakyReLU and to avoid neuron inactivity, make LeakyReLU better option to go with. The second block comprises four layers: a Conv2D layer, a Dropout layer with a dropout rate of 0.2, a LeakyReLU activation function, and a Batch Normalization (BNorm) layer. The third block involves three Conv2D layers. The outputs of the initial three convolutional layers are subsequently inputted into corresponding fourth block having deconvolutional layers, facilitating the reconstruction of the denoised image refer to Table 2.

The Summary of four blocks is as follows: [Ioffe & Szegedy, 2015] [Maas et al., 2013]

1.**Block 1**: Consists of a Conv2D layer and an activation function called LeakyReLU. [Maas et al., 2013]

2. **Block 2**: The composition includes the following layers: Batch Normalization (BNorm), LeakyReLU activation, dropout layer (dropout rate = 0.2), and Conv2D layer. [Ioffe & Szegedy, 2015] [Maas et al., 2013]

3. **Block 3**: Composed of three Conv2D layers.

4. **Block 4**: The output from the previous blocks is processed by deconvolutional layers to reconstruct the denoised image.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

For more clarity refer to Table 2

Block	Layer Type	Number of Filters	Kernel Size	Stride	Padding
Block 1	Conv2D	64	3x3	1	Default/Same
	LeakyReLU	-	-	-	-
Block 2	Conv2D	64	3x3	2	Default/Same
	Droupout	-	-	-	-
	LeakyReLU	-	-	-	-
	BNorm	-	-	-	-
Block 3	Conv2D	128	5x5	2	Default/Same
	Conv2D	128	3x3	1	Default/Same
	Conv2D	256	5x5	2	Default/Same
Block 4	Deconv2D	512	3x3	1	Default/Same
	Deconv2D	512	5x5	2	Default/Same
	Deconv2D	256	7x7	1	Default/Same
	Deconv2D	128	9x9	2	Default/Same
	Deconv2D	128	7x7	1	Default/Same
	Deconv2D	64	5x5	2	Default/Same
	Deconv2D	64	3x3	1	Default/Same

Table 2:Layer wise configuration of proposed Model

The ResNet denoiser as shown in Figure5(b) benefits from residual connections, which help optimize network learning by permitting residual mappings between each layer's input and output to be learned by the model, improving denoising performance. The image size increases progressively through five consecutive convolution operations and subsequent deconvolution steps, from 16 pixels to 500 pixels.

Each convolutional block contains layers with varying filter sizes and kernel dimensions, with dilated convolution kernels: 3×3, 5×5, 7×7, 9×9, 7×7, 5×5, and 3×3. Padding is set to default, and the stride alternates between 1 and 2. Orthogonal initialization is used for kernel weights, and zero padding ensures feature map consistency before deconvolution.

Dropout layers in each block help combat overfitting by randomly masking feature activations, encouraging the model to learn more robust representations. Batch normalization is incorporated into the final product of each deconvolutional block to stabilize training and improve performance. ConvTranspose layers in the deconvolutional blocks have increasing filter sizes and kernel dimensions. The final output is compared to the input image to assess the denoising performance.The ConvResNet model effectively eliminates common noise varitites, including Gaussian noise, and Salt and Pepper and Speckle noise. [Ioffe & Szegedy, 2015]

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

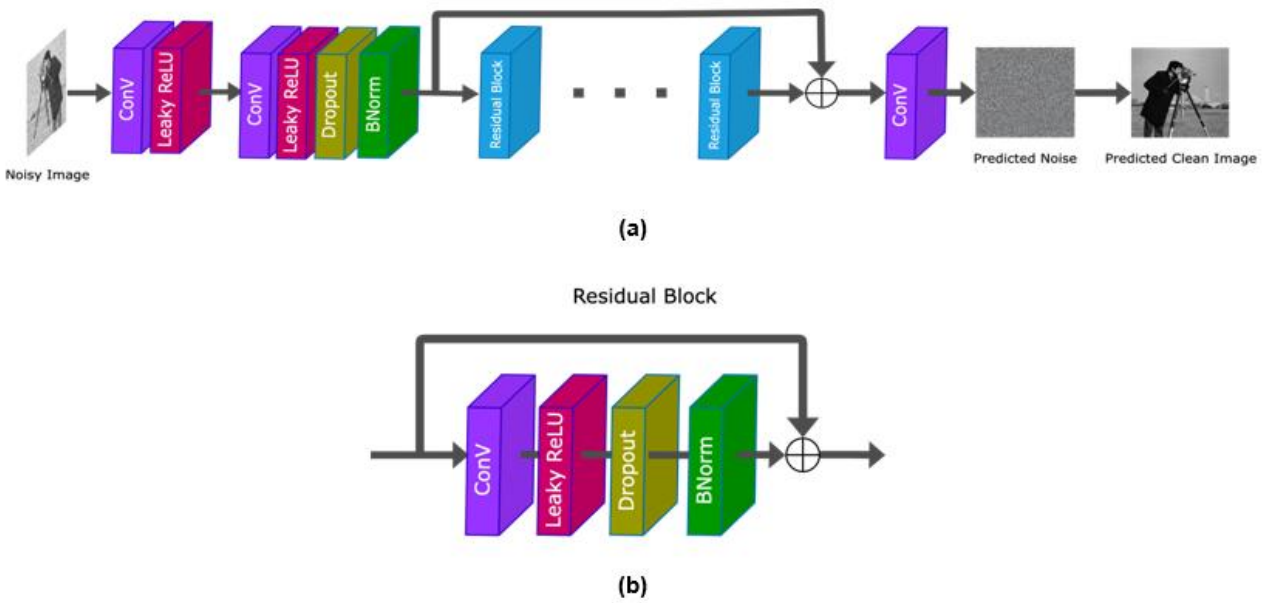


Figure 5. (a) Overview of model architecture (b) Residual Block

2.4. Batch Normalization [Ioffe & Szegedy, 2015]

In deep learning models, it is very popular technique called Batch normalization. Deep learning model captures mapping function to convert the input feature of any input sample. It is essential in accelerating the training phase and boosting the generalization of learning neural networks. By removing the mean and dividing by the mini batch's standard deviation, the batch normalization procedure normalizes a layer's output. This aids in keeping the inputs to the following layers at zero mean and unit variance. Equations (11) and (12) yield the batch's mean $\bar{x}$ and standard deviation (σ). [Ioffe & Szegedy, 2015]

$$\overline{x} = \frac{1}{m}\sum_{i=0}^{m} xi \quad (11)$$

$$\sigma = \sum_{i=0}^{m}(xi - \overline{x})^2 \quad (12)$$

m is the batch size, xi is the i^th input to the layer, and $\bar{x}$ and (σ) are the mean and variance of the batch. To normalize the input batch, equation (13) is applied:

$$xi = \frac{(xi - \overline{x})}{\sigma + \epsilon} \quad (13)$$

In this equation, ∈ is a small constant (typically 10^{-5}) added for numerical stability. After normalization, the batch is scaled and shifted using equation (14):

$$y_i = \gamma * x_i + \beta \quad (14)$$

The parameters γ and β can be adjusted by the network during the training process this fine-tuning helps the network achieve the best performance for each layer, leading to better overall results. The integration of batch normalization accelerates convergence, and this also enhances stability and improves generalization. This ensures efficient gradient propagation and reduces issues with vanishing or exploding gradients. [Ioffe & Szegedy, 2015]

2.5. Training

This model uses the binary cross-entropy loss function, amongst the popular choice for binary classification problems. The Adam optimizer was utilised during the training process with a learning rate of 0.00001 to minimize the loss function and adjust the model's parameters, thereby enhancing the model's performance. The optimizer modified the model's parameters over 50 training epochs. The training loss was monitored and displayed on a graph for these 50 epochs to evaluate the model's performance over time. Graph illustrating the training loss versus epochs is attached in section 5, providing insight into how the loss decreased over time and the model's learning effectiveness from the training data. [Kingma & Ba, 2017]

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A loss function calculates the difference between the actual target values and the model's predicted output. It helps the model learn by providing a measure of its performance.

$$\textbf{Loss} = \frac{1}{2N}\sum_{i=1}^{N}\|\mathbf{f}[(\mathbf{y_i}, \Theta) - (\mathbf{y_i} - \mathbf{x_i})]^2\| \quad \textbf{(15)}$$

Here, y_i represents the true target values, x_i represents the forecasted output of the model, and N is sample number in the dataset. Θ (The loss function) is established using the Adam solver (Kingma & Ba, 2017), and the remaining hyperparameters of Adam adopt their default values as shown in equation (15) and Table 3 (Summary of Hyperparameters used in the Training Process)

Hyperparameter	Value
Batch Size	32
Initial Learning Rate	0.00001
Learning Rate Decay	Step Decay
Decay Factor	0.1
Decay Step	Every 10 epochs
Optimization Algorithm	Adam
Loss Function	Binary Cross-Entropy
Number of Epochs	50

Table 3: Summary of Hyperparameters used in the Training Process

3. EXPERIMENT RESULTS AND DISCUSSION

Images were divided into 2x2 pixel blocks and scrambled. Two arbitrary phase masks were generated using a fractional Fourier transform (α = 0.5, β = 0.5). Figure 6 shows:

- (a) Original image
- (b) Scrambled image
- (c) Encrypted image
- (d) Decrypted noisy image

(e) Denoised The proposed Deep Conv ResNet model was implemented on a Windows 10 system with a 3.3 GHz Intel i5-CPU and 8 GB RAM. The model was trained and tested using Python 3. Three datasets were used: Waterloo Exploration Database (Ma et al., 2017) with 4744 images, McMaster's dataset (Q. Zhang et al., 2023) with 18 cropped images, and BSMCD68 dataset (Yaman et al., 2020) with 68 images. Each image was 500x500 pixels. The datasets links are available in the data availability section.Datasets were divided 80:20 between testing and training. To ensure uniformity, pixel values were standardized between 0 and 1. Multiple experiments were conducted to validate the approach. The model was tested on grayscale images "Camera Man" and "Bell Pepper," each 512x512 pixels image using Deep Conv ResNet

Noise (salt-and-pepper, speckle, AWGN) was added to encrypted images. Decryption was performed without noise elimination. Figure 7 illustrates the attacks:

(a, b) Salt and pepper noise
(c, d) Speckle noise
(e, f) Gaussian noise (variance 0.5)
(g, h) Gaussian noise (variance 0.10)
(i, j) Gaussian noise (variance 0.15)

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

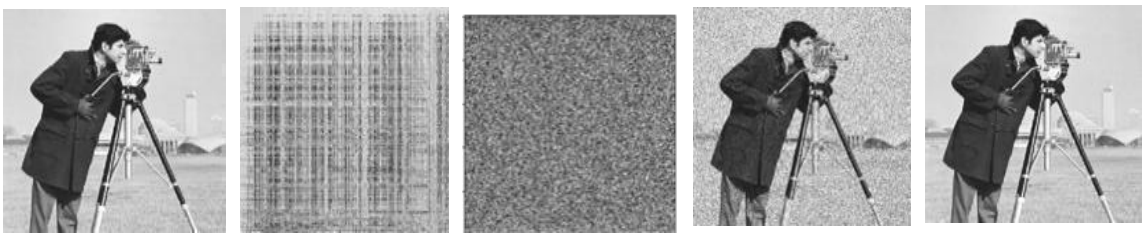


Figure 6. (a) Original image (b) Scrambled image (c) Encrypted image (d) Decrypted noisy image (e) denoised image using deep ConvResnet denoiser.

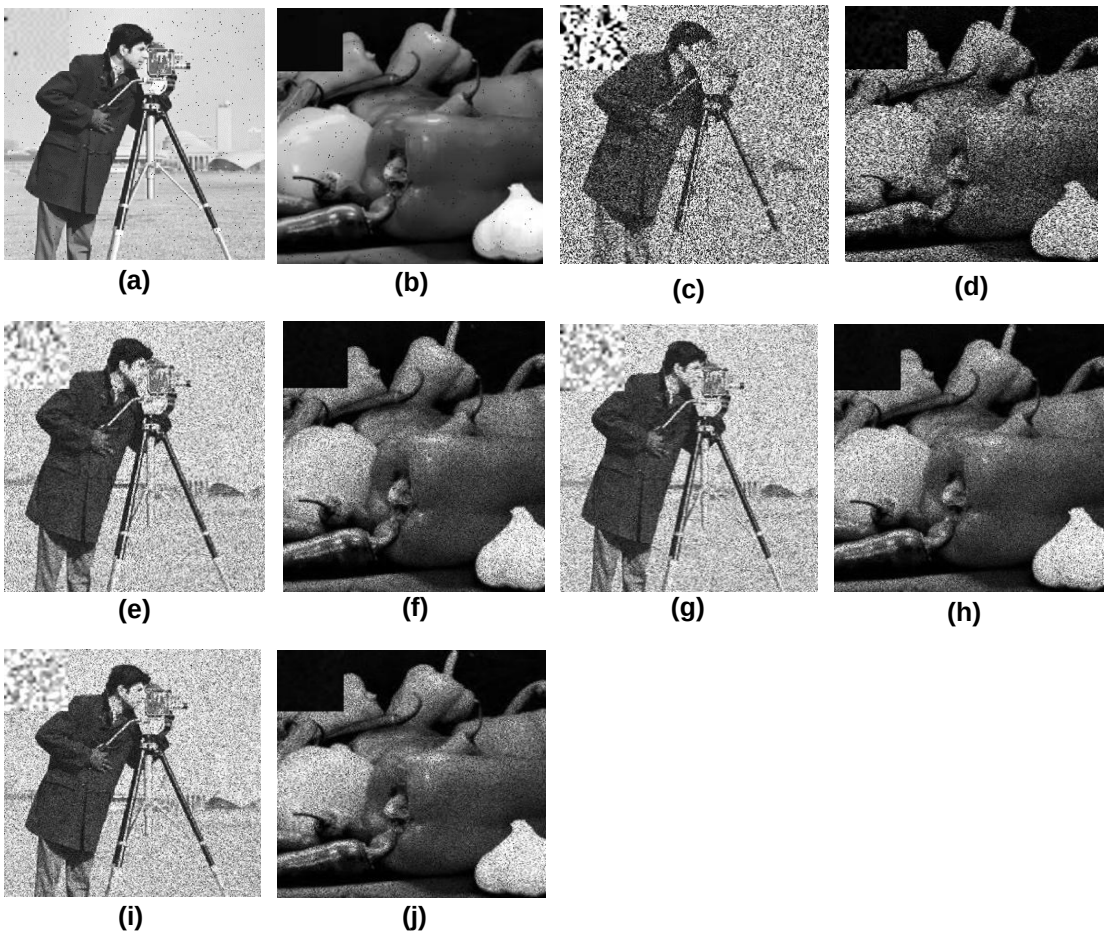


Figure 7: Decrypted images with different noise attacks: (a)-(b) salt and pepper noise attack, (c)-(d) speckle noise attack, (e)-(f) AWGN attack with variance 0.05 (g)-(h) AWGN with variance 0.10, (i)-(j) AWGN with variance 0.15

3.1. Simulation Parameters (including noise analysis)

3.1.1.1. SSIM

By contrasting the luminance, contrast, and structural details of two images, the Structural Similarity Index (SSIM) calculates how similar they are. It evaluates the performance of image processing algorithms or compares the quality of compressed images with their original counterparts (Singh et al., 2015). The formula for SSIM is given by equation (16):

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$SSIM_{(x,y)} = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (16)$$

Where:
x is the original image and y is the encrypted image.
μx and μy are the pixel sample means of the x and y images, respectively.
σxy is the covariance of x and y.
σ2x and σ2y are the variances of the x and y images, respectively.
C1 = (k1 L)2 and c2 = (k2 L)2, where L is the dynamic range of the image (2bit -1).
The SSIM index ranges from -1 to 1, with 1 indicating identical images.

3.1.1.2.PSNR

PSNR (Peak Signal to Noise Ratio) is a widely used metric to assess the quality of both the original and reconstructed images. The calculation determines the ratio of a signal's maximal strength to the power of the noise that degrades the representation's fidelity (Khurana & Singh, 2018a). The formula for PSNR is:

$$PSNR = 10 \times log_{10}(\frac{255^2}{MSE}) \quad (17)$$

The maximum pixel value of the image is 255, while the mean squared error (MSE) between the original and reconstructed images is 255. Both PSNR and SSIM (Structural Similarity Index Measure) values are used to evaluate the quality of reconstructed pictures.
Results for different types of noise like; AWGN (Additive White Gaussian Noise), SNP (Salt and Pepper), Speckle noise are presented in Table 4, Table 5, and Table 6). The suggested method achieved high-quality reconstruction despite various noise attacks on different datasets, such as Waterloo, McMaster, and MCBSD68.

Noise	PSNR VALUES (dB)			SSIM VALUES		
	Variance	Camera Man	Bell Pepper	Variance	Camera Man	Bell Pepper
Gaussian	0.05	48.61985	41.66349	0.05	0.7765	0.9232
	0.1	36.81562	39.87612	0.1	0.749	0.8937
	0.15	35.27775	38.06106	0.15	0.7029	0.8683

Table 4(a). Average PSNR and mean SSIM for Gaussian Noise for Waterloo dataset.

Noise type	PSNR VALUES (dB)		SSIM VALUES	
	Camera Man	Bell Pepper	Camera Man	Bell Pepper
Salt and Pepper	42.40102	44.03456	0.8692	0.9632
Speckle	31.89734	42.40102	0.7138	0.8692

Table 4(b). Average PSNR and mean SSIM for SNP & Speckle Noise for Waterloo dataset.

Noise	PSNR VALUES (dB)			SSIM VALUES		
	Variance	Camera Man	Bell Pepper	Variance	Camera Man	Bell Pepper
Gaussian	0.05	21.990559	28.4518295	0.05	0.608036068	0.722578177
	0.1	20.97864054	27.6931722	0.1	0.563752792	0.69619915
	0.15	20.15876493	26.9333239	0.15	0.518801557	0.67715711

Table 5(a). Average PSNR and mean SSIM for Gaussian Noise of McMaster dataset.

Noise type	PSNR VALUES (dB)		SSIM VALUES	
	Camera Man	Bell Pepper	Camera Man	Bell Pepper
Salt and Pepper	23.14534359	29.2553896	0.721203065	0.75838368
Speckle	15.34813799	23.1827596	0.323107252	0.51875745

Table 5(b). Average PSNR and mean SSIM for SNP & Speckle Noise for McMaster dataset.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Noise	PSNR VALUES (dB)			SSIM VALUES		
	Variance	Camera Man	Bell Pepper	Variance	Camera Man	Bell Pepper
Gaussian	0.05	21.99056	28.45183	0.05	0.608036	0.722578
	0.1	20.97864	27.69317	0.1	0.563753	0.696199
	0.15	20.15876	26.93332	0.15	0.518802	0.677157

Table 6(a). Average PSNR and mean SSIM for Gaussian Noise for MCBSD68 dataset

Noise type	PSNR VALUES (dB)		SSIM VALUES	
	Camera Man	Bell Pepper	Camera Man	Bell Pepper
Salt and Pepper	23.14534	29.25539	0.721203	0.758384
Speckle	15.34814	23.18276	0.323107	0.518757

Table 6(b). Average PSNR and mean SSIM for SNP & Speckle Noise for MCBSD68 dataset.

3.2. Robustness Against Attacks

Various kinds of attacks u=were used to check the robustness of the model such as noise, blur, and occlusion were used. Noise attacks involved adding random noise to an image, distorting it and making it challenging for the recognition system to identify objects correctly (Zhu et al., 2022). Figure 7 shows the noisy images processed through the proposed algorithm, which includes a denoiser. Also Figures 8(a) & 8(c) show inputs to the decryption algorithm, were as Figures 8(b) & 8(d) display the outputs after applying the deep ConvResNet denoiser against salt and pepper noise. Similarly, Figures 8(e) & 8(g) and 8(f) & 8(h) demonstrate the denoising performance against speckle noise. Whereas Figures 8(i) & 8(k) and 8(j) & 8(l) show results for AWGN noise with variance 0.5, and Figures 8(m) & 8(o) and 8(n) & 8(p) shows results for AWGN noise with variance 0.10. The proposed algorithm effectively denoises and recovers clean images, as seen in Figure 8.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

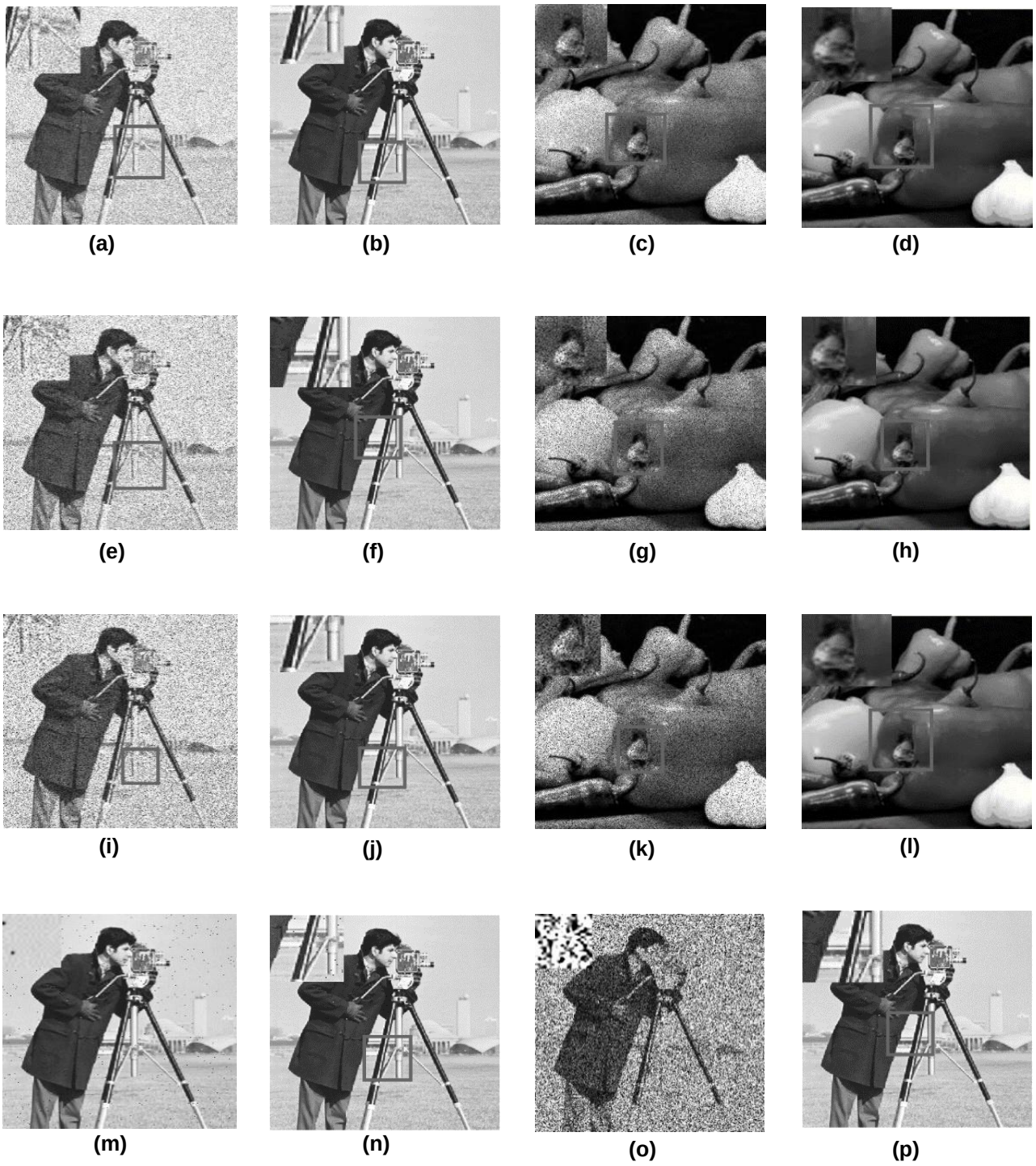


Figure 8. Figures (a), (c), (e), (g), (i), (k), (m), and (o) represent the images without utilizing the deep ConvResNet denoiser, whereas subfigures (b), (d), (f), (h), (j), (l), (n), and (p) display the reconstructed images employing the deep ConvResNet denoiser.

3.3. Occlusion Attack

An occlusion attack tests a recognition system's ability to identify objects that are partially visible, either due to obstruction by other objects or occlusion from shadows. This type of attack plays a vital role for assessing the robustness and accuracy of image recognition systems. Figure 9 shows the results of occlusion attacks on

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

images with 40% and 60% occlusion. Additionally, Figure 10 presents the outcomes of the blur attack.

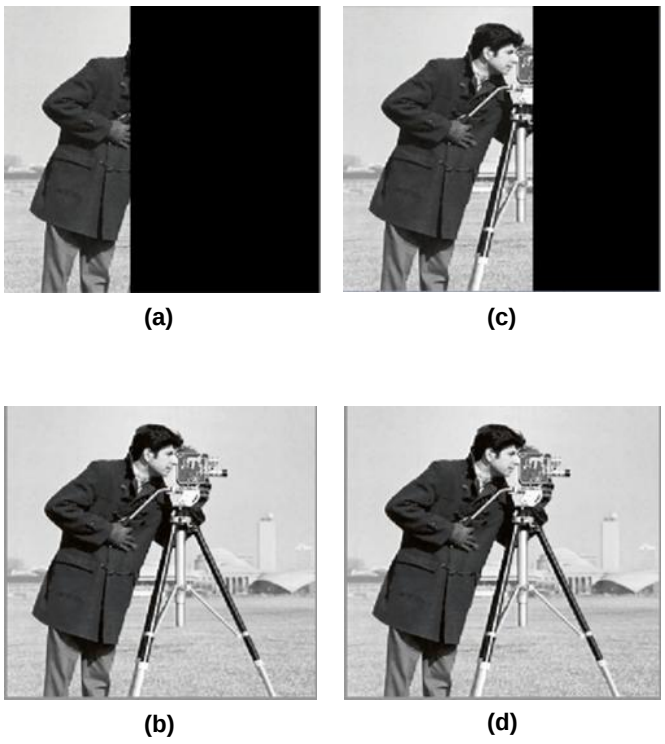


Figure 9. Decrypted images with occlusion attack: (a) 60 % occluded, (b) recovered image, (c) 40 % occluded, (d) recovered image.

Figure 10. Decrypted images with blur attack: (a) input image, (b) blur attack, (c) recovered image.

3.4. Comparative Analysis

Figure 10 shows the results obtained with a blur attack. The suggested approach is contrasted with a number of well-known denoising models: BM3D (Burger et al., 2012), MLP (Huang et al., 2023), CNN (Chen et al., 2019), and ConvLSTM (Piriyatharawet et al., 2018). Figure 11 provides a visual representation of the comparative results from different models verified for AWGN noise with variance 0.10 for Camera Man and Bell Pepper images. The images with AWGN of variances 0.05, 0.10, and 0.15 were passed through various denoising models, and results were compared for PSNR & SSIM values.

Output results are shown in Table 4 and Table 5 for PSNR and SSIM, respectively. The proposed deep ConvResNet Denoiser achieved an average increase of 66.94% and 86.65% in PSNR, as shown in Table 7. In comparison, the BM3D method resulted in an increment of 11.06%, MLP achieved 21.06%, CNN saw a 21.64%

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

increase, and CLSTM achieved 15.15%. While PSNR values of the noisy decrypted pictures derived from the suggested model are the highest in the first column of Table 4, PSNR alone does not align with human visual perception. Therefore, image quality should also be assessed by considering SSIM values.

Figure 8 shows images without the deep ConvResNet denoiser in subfigures (a), (c), (e), (g), (i), (k), (m), and (o), while subfigures (b), (d), (f), (h), (j), (l), (n), and (p) display the reconstructed images using the proposed denoiser.

Figure 9 illustrates decrypted images with occlusion attacks: (a) 60% occluded, (b) recovered image, (c) 40% occluded, (d) recovered image. Figure 10 presents decrypted images with blur attacks: (a) input image, (b) blur attack, (c) recovered image.

Table 8 shows that the mean SSIM values improved by 14.89% with the proposed deep ConvResNet Denoiser. In comparison, the BM3D method increased by 11.89%, MLP by 7.33%, CNN by 13.87%, and CLSTM by 13.99%. These findings show that the suggested approach provides better picture reconstruction and performs better than alternatives in terms of both PSNR and SSIM.

PSNR Values						
Variance Gaussian (AWGN)	0.05		0.10		0.15	
	Camera Man	Bell Pepper	Camera Man	Bell Pepper	Camera Man	Bell Pepper
Noisy Image	**27.5267**	**28.7489**	**24.6537**	**19.2429**	**20.1558**	**16.1216**
BM3D(Burger et al., 2012)	28.0342	26.7814	25.4293	20.1413	26.87 43	17.9534
MLP(Huang et al., 2023)	26.4564	25.3193	27.4173	22.3447	23.6844	18.2464
CNN(Chen et al., 2019)	29.6752	32.5496	28.76221	31.1532	27.56074	29.7352
CLSTM(Piriyatharawet et al., 2018)	31.2512	28.9111	29.5134	24.5156	22.5363	17.5214
Deep ConvResNet	**48.6198**	**41.6634**	**36.8156**	**39.8761**	**35.2777**	**38.0610**

Table 7. PSNR values of the decrypted image by different denoising models

SSIM Values			
Variance	**0.05**	**0.10**	0.15
C.man noisy image	**0.2314**	**0.4217**	0.4258
BM3D(Burger et al., 2012)	0.8391	0.7862	0.7380
MLP(Huang et al., 2023)	0.7011	0.6579	0.5114
CNN(Chen et al., 2019)	0.8885	0.8729	0.8135
CLSTM(Piriyatharawet et al., 2018)	0.8971	0.7923	0.8895
DeepConvResNet	0.9232	0.8937	0.8683

Table 8. SSIM values of the decrypted image by different denoising models

3.5. Epoch

A full cycle through the training data constitutes an epoch. The number of epochs functions as a hyperparameter to determine how often the entire dataset will be traversed. Model training depends on Epoch because it determines how well the model detects essential patterns.

The graph in Figure 11 shows the relationship between average PSNR (Peak Signal-to-Noise Ratio) and epochs. The vertical axis contains average PSNR data which indicates image denoising quality through the training process shown through the horizontal axis that tracks epochs.

The model begins with low average PSNR values at the start of image denoising before adjusting to the process. Each successive epoch leads to an increasingly better performance of the proposed model through an upward

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

trend in average PSNR measure. The model demonstrates enhanced capability to decrease noise and maintain vital image elements as it progresses through its training process. The model reached a loss value of 0.5439 during the 50th epoch completion. The model learning process evaluation and the comparison between predicted and actual values became possible through this method during training.

The training process of McMaster data appears in Graph 12(a) while Graph 12(b) displays MCBSD68 data loss. The accuracy assessment of the model for training and validation data appears in graph 12(c) to evaluate the model's ability to generalize. The Model's loss during training (on the Waterloo dataset) appears in graph 12(d) to show how the model converged and performed.

The presented graphs display significant metrics and performance indicators throughout the training process for various datasets. The default acceptance of the results helps examine model performance and determine better adjustments to optimize end results.

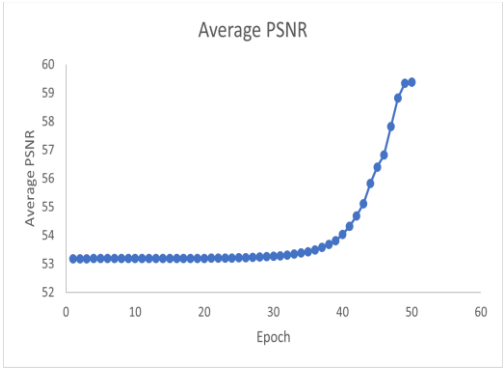
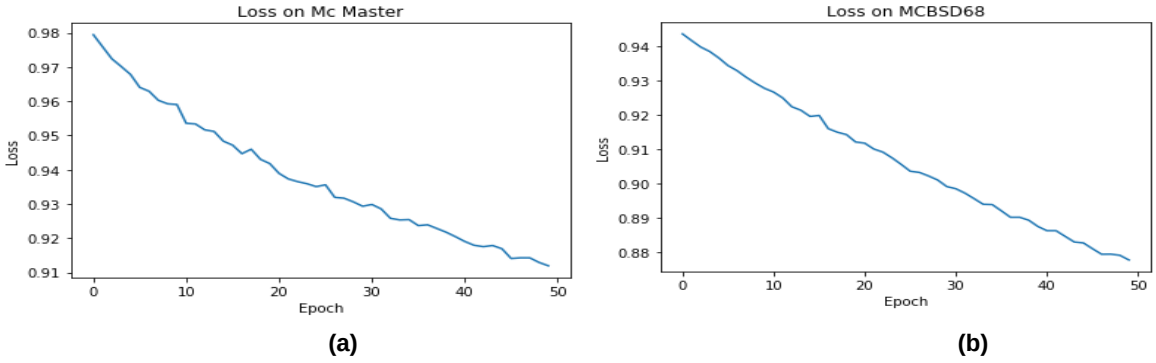


Figure 11. Graph of average PSNR versus Epoch



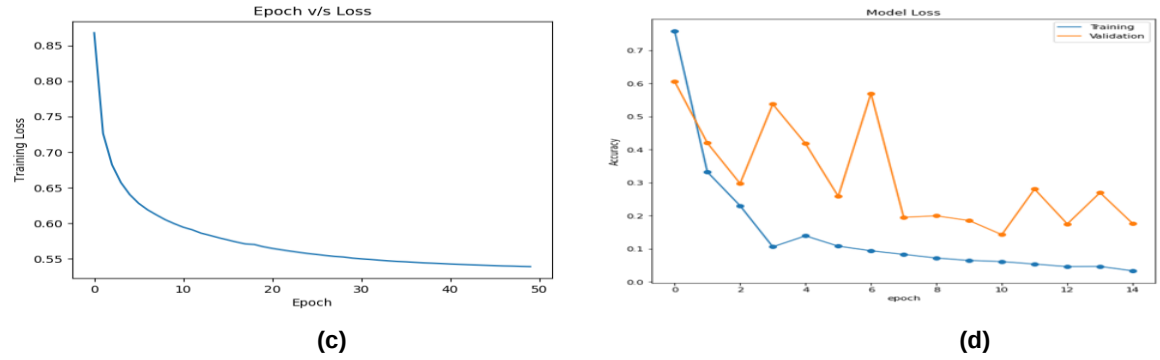


Figure 12. Graphs: (a) loss in training data versus Epoch on McMaster, (b) loss in training data versus Epoch on MCBSD68 dataset, (c) accuracy of training and validation data (d) Model loss of waterloo dataset of accuracy vs epoch.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4. CONCLUSION

This work introduced an enhanced deep convolutional residual network (ConvResNet) designed to improve the quality of decrypted images that have been subjected to various noise attacks. Through extensive testing on benchmark datasets and diverse noise models, the proposed architecture demonstrated superior performance compared to traditional and modern denoising techniques, including BM3D, MLP, CNN, and ConvLSTM. The model consistently achieved higher PSNR and SSIM scores, validating its effectiveness in preserving structural details and minimizing distortion. Training curves further confirmed the model's stability and convergence, indicating its robustness across multiple experimental settings.

5. FUTURE WORKS

In future investigations, the framework will be extended to address colored encrypted images and encrypted video streams, thus broadening its applicability. Furthermore, optimizing the model for real-time performance on low-power devices such as IoT nodes and embedded platforms will be explored. Incorporating adaptive noise estimation and dynamic residual learning techniques could further enhance noise resilience, offering stronger performance across a broader range of transmission and encryption scenarios.

INFORMED CONSENT

All participants have given their informed consent to the author.

CONFLICT OF INTEREST

The author discloses that no competing interests are present

DATA AVAILABLITY

The datasets which are used in this article are freely available for download on the following links:
1.https://ece.uwaterloo.ca/~k29ma/exploration
2.https://www.kaggle.com/code/mpwolke/berkeley-segmentation-dataset-68
3.https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ respectively

REFERENCES

[1] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. In ICML.
[2] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In ICML.
[3] Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980
[4] Refregier, P., & Javidi, B. (1995). Optical image encryption based on input plane and Fourier plane random encoding. Optics Letters, 20(7), 767–769. https://doi.org/10.1364/OL.20.000767
[5] Unnikrishnan, G., Joseph, J., & Singh, K. (2000). Optical encryption by double-random phase encoding in the fractional Fourier domain. Optics Letters, 25(12), 887–889. https://doi.org/10.1364/OL.25.000887
[6] Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing, 26(7), 3142–3155. https://doi.org/10.1109/TIP.2017.2662206
[7] Zhang, K., Zuo, W., & Zhang, L. (2018). FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. IEEE Transactions on Image Processing, 27(9), 4608–4622. https://doi.org/10.1109/TIP.2018.2839891
[8] Burger, H. C., Schuler, C. J., & Harmeling, S. (2012). Image denoising: Can plain neural networks compete with BM3D? 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2392–2399. https://doi.org/10.1109/CVPR.2012.6247952

[9] Dong, W., Wang, P., Yin, W., Shi, G., Wu, F., & Lu, X. (2019). Denoising Prior Driven Deep Neural Network for Image Restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(10), 2305–2318. https://doi.org/10.1109/TPAMI.2018.2873610

[10] Li, J., Zhang, Z., Liu, X., Feng, C., Wang, X., Lei, L., & Zuo, W. (2023). Spatially Adaptive Self-Supervised Learning for Real-World Image Denoising (arXiv:2303.14934). arXiv. https://doi.org/10.48550/arXiv.2303.14934

[11] Ren, H., El-khamy, M., & Lee, J. (2019). DN-ResNet: Efficient Deep Residual Network for Image Denoising. In C. V. Jawahar, H. Li, G. Mori, & K. Schindler (Eds.), Computer Vision – ACCV 2018 (pp. 215–230). Springer International Publishing. https://doi.org/10.1007/978-3-030-20873-8_14

[12] Zhang, J., Zhu, Y., Li, W., Fu, W., & Cao, L. (2021). DRNet: A Deep Neural Network With Multi-Layer Residual Blocks Improves Image Denoising. IEEE Access, 9, 79936–79946. https://doi.org/10.1109/ACCESS.2021.3084951

[13] Chen, J., Li, X.-W., & Wang, Q.-H. (2019). Deep Learning for Improving the Robustness of Image Encryption. IEEE Access, 7, 181083–181091. https://doi.org/10.1109/ACCESS.2019.2959031

[14] Anshula, & Singh, H. (2021). Security-enrichment of an asymmetric optical image encryption-based devil's vortex Fresnel lens phase mask and lower upper decomposition with partial pivoting in gyrator transform domain. *Optical and Quantum Electronics*, *53*(4), 204. https://doi.org/10.1007/s11082-021-02854-7

[15] Arora, M., & Khurana, M. (2020). Secure image encryption technique based on jigsaw transform and chaotic scrambling using digital image watermarking. *Optical and Quantum Electronics*, *52*(2), 59. https://doi.org/10.1007/s11082-019-2130-3

[16] Bhat, O. V. (2021). *A Review on Image Denoising Techniques*. *8*, 1779–1784. https://www.irjet.net/archives/V8/i9/IRJET-V8I9254.pdf

[17] Buades, A., Coll, B., & Morel, J.-M. (2005). A non-local algorithm for image denoising. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *2*, 60–65 vol. 2. https://doi.org/10.1109/CVPR.2005.38

[18] Burger, H. C., Schuler, C. J., & Harmeling, S. (2012). Image denoising: Can plain neural networks compete with BM3D? *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2392–2399. https://doi.org/10.1109/CVPR.2012.6247952

[19] Chen, J., Li, X.-W., & Wang, Q.-H. (2019). Deep Learning for Improving the Robustness of Image Encryption. *IEEE Access*, *7*, 181083–181091. https://doi.org/10.1109/ACCESS.2019.2959031

[20] Ding, Y., Wu, G., Chen, D., Zhang, N., Gong, L., Cao, M., & Qin, Z. (2021). DeepEDN: A Deep-Learning-Based Image Encryption and Decryption Network for Internet of Medical Things. *IEEE Internet of Things Journal*, *8*(3), 1504–1518. https://doi.org/10.1109/JIOT.2020.3012452

[21] Dong, W., Wang, P., Yin, W., Shi, G., Wu, F., & Lu, X. (2019). Denoising Prior Driven Deep Neural Network for Image Restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(10), 2305–2318. https://doi.org/10.1109/TPAMI.2018.2873610

[22] Huang, Z., Zhu, Z., Zhang, Y., Wang, Z., Xu, B., Liu, J., Li, S., & Fang, H. (2023). MD3: Model-Driven Deep Remotely Sensed Image Denoising. *Remote Sensing*, *15*(2), Article 2. https://doi.org/10.3390/rs15020445

[23] Jing, J., Xia, W., Hou, M., Chen, H., Liu, Y., Zhou, J., & Zhang, Y. (2022). Training low dose CT denoising network without high quality reference data. *Physics in Medicine & Biology*, *67*(8), 084002. https://doi.org/10.1088/1361-6560/ac5f70

[24] Khurana, M., & Singh, H. (2018a). Data Computation and Secure Encryption Based on Gyrator Transform using Singular Value Decomposition and Randomization. *Procedia Computer Science*, *132*, 1636–1645. https://doi.org/10.1016/j.procs.2018.05.129

[25] Khurana, M., & Singh, H. (2018b). Spiral-Phase Masked Optical Image Health Care Encryption System for Medical Images Based on Fast Walsh-Hadamard Transform for Security Enhancement. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, *13*(4), 98–117. https://doi.org/10.4018/IJHISI.2018100107

[26] Khurana, M., & Singh, H. (2020). Two level phase retrieval in fractional Hartley domain for secure image encryption and authentication using digital signatures. *Multimedia Tools and Applications*, *79*(19), 13967–13986. https://doi.org/10.1007/s11042-020-08658-3

[27] Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. https://doi.org/10.48550/arXiv.1412.6980
[28] Li, J., Zhang, Z., Liu, X., Feng, C., Wang, X., Lei, L., & Zuo, W. (2023). *Spatially Adaptive Self-Supervised Learning for Real-World Image Denoising* (arXiv:2303.14934). arXiv. https://doi.org/10.48550/arXiv.2303.14934
[29] Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., & Zhang, L. (2017). Waterloo Exploration Database: New Challenges for Image Quality Assessment Models. *IEEE Transactions on Image Processing*, *26*(2), 1004–1016. https://doi.org/10.1109/TIP.2016.2631888
[30] Maharjan, P., Li, L., Li, Z., Xu, N., Ma, C., & Li, Y. (2019). Improving Extreme Low-Light Image Denoising via Residual Learning. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 916–921. https://doi.org/10.1109/ICME.2019.00162
[31] Piriyatharawet, T., Kumwilaisak, W., & Lasang, P. (2018). Image Denoising with Deep Convolutional and Multi-directional LSTM Networks under Poisson Noise Environments. *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, 90–95. https://doi.org/10.1109/ISCIT.2018.8588010
[32] Raj, A., Bresler, Y., & Li, B. (2020). *Improving Robustness of Deep-Learning-Based Image Reconstruction* (arXiv:2002.11821). arXiv. https://doi.org/10.48550/arXiv.2002.11821
[33] Refregier, P., & Javidi, B. (1995). Optical image encryption based on input plane and Fourier plane random encoding. *Optics Letters*, *20*(7), 767–769. https://doi.org/10.1364/OL.20.000767
[34] Ren, H., El-khamy, M., & Lee, J. (2019). DN-ResNet: Efficient Deep Residual Network for Image Denoising. In C. V. Jawahar, H. Li, G. Mori, & K. Schindler (Eds.), *Computer Vision – ACCV 2018* (pp. 215–230). Springer International Publishing. https://doi.org/10.1007/978-3-030-20873-8_14
[35] Sharma, B. (2013). SECURITY ARCHITECTURE OF CLOUD COMPUTING BASED ON ELLIPTIC CURVE CRYPTOGRAPHY (ECC). *International Journal of Advances in Engineering Sciences*, *3*, 58–61.
[36] Sharma, B., & Singh, J. (2022). *Chaos Based Image Encryption Techniques: A Review*. *09*, 827–835. https://doi.org/10.13140/RG.2.2.26235.39204
[37] Singh, H., Ahmed, A. S., Melandsø, F., & Habib, A. (2023). Ultrasonic image denoising using machine learning in point contact excitation and detection method. *Ultrasonics*, *127*, 106834. https://doi.org/10.1016/j.ultras.2022.106834
[38] Singh, H., Yadav, A. K., Vashisth, S., & Singh, K. (2015). Double phase-image encryption using gyrator transforms, and structured phase mask in the frequency plane. *Optics and Lasers in Engineering*, *67*, 145–156. https://doi.org/10.1016/j.optlaseng.2014.10.011
[39] Tian, C., Xu, Y., & Zuo, W. (2020). Image denoising using deep CNN with batch renormalization. *Neural Networks*, *121*, 461–473. https://doi.org/10.1016/j.neunet.2019.08.022
[40] Tian, C., Zheng, M., Zuo, W., Zhang, B., Zhang, Y., & Zhang, D. (2023). Multi-stage image denoising with the wavelet transform. *Pattern Recognition*, *134*, 109050. https://doi.org/10.1016/j.patcog.2022.109050
[41] Unnikrishnan, G., Joseph, J., & Singh, K. (2000). Optical encryption by double-random phase encoding in the fractional Fourier domain. *Optics Letters*, *25*(12), 887–889. https://doi.org/10.1364/OL.25.000887
[42] Sharma, B., & Singh, J. (2022). Chaos based image encryption techniques: A review. International Journal, 9, 827–835. https://doi.org/10.13140/RG.2.2.26235.39204
[43] Wei, F., Zhenhao, Z., Zhou, H., Tao, Z., Jun, S., & Xiaojun, W. (2023). Efficient automatically evolving convolutional neural network for image denoising. *Memetic Computing*, *15*(2), 219–235. https://doi.org/10.1007/s12293-022-00385-6
[44] Wei, X., Xiao, J., & Gong, Y. (2023). Blind Hyperspectral Image Denoising with Degradation Information Learning. *Remote Sensing*, *15*(2), Article 2. https://doi.org/10.3390/rs15020490
[45] Xu, J., Yuan, M., Yan, D.-M., & Wu, T. (2023). Deep unfolding multi-scale regularizer network for image denoising. *Computational Visual Media*, *9*(2), 335–350. https://doi.org/10.1007/s41095-022-0277-5
[46] Yaman, B., Hosseini, S. A. H., & Akçakaya, M. (2020). *Noise2Inpaint: Learning Referenceless Denoising by Inpainting Unrolling*.
[47] Yan, R., Liu, Y., Liu, Y., Wang, L., Zhao, R., Bai, Y., & Gui, Z. (2023). Image Denoising for Low-Dose CT via Convolutional Dictionary Learning and Neural Network. *IEEE Transactions on Computational Imaging*, *9*,

83–93. https://doi.org/10.1109/TCI.2023.3241546

[48] Yuan, Q., Zhang, Q., Li, J., Shen, H., & Zhang, L. (2019). Hyperspectral Image Denoising Employing a Spatial–Spectral Deep Residual Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(2), 1205–1218. https://doi.org/10.1109/TGRS.2018.2865197

[49] Zhang, J., Zhu, Y., Li, W., Fu, W., & Cao, L. (2021). DRNet: A Deep Neural Network With Multi-Layer Residual Blocks Improves Image Denoising. *IEEE Access*, *9*, 79936–79946. https://doi.org/10.1109/ACCESS.2021.3084951

[50] Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, *26*(7), 3142–3155. https://doi.org/10.1109/TIP.2017.2662206

[51] Zhang, K., Zuo, W., Gu, S., & Zhang, L. (2017). Learning Deep CNN Denoiser Prior for Image Restoration. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2808–2817. https://doi.org/10.1109/CVPR.2017.300

[52] Zhang, K., Zuo, W., & Zhang, L. (2018). FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Transactions on Image Processing*, *27*(9), 4608–4622. https://doi.org/10.1109/TIP.2018.2839891

[53] Zhang, Q., Xiao, J., Tian, C., Chun-Wei Lin, J., & Zhang, S. (2023). A robust deformed convolutional neural network (CNN) for image denoising. *CAAI Transactions on Intelligence Technology*, *8*(2), 331–342. https://doi.org/10.1049/cit2.12110

[54] Zhu, Y., Abdalla, A., Tang, Z., & Cen, H. (2022). Improving rice nitrogen stress diagnosis by denoising strips in hyperspectral images via deep learning. *Biosystems Engineering*, *219*, 165–176. https://doi.org/10.1016/j.biosystemseng.2022.05.001

[55] Sharma, B. (2013). Security architecture of cloud computing based on elliptic curve cryptography (ECC). International Journal of Advances in Engineering Sciences, 3, 58–61.

[56] Sharma, B., Khurana, M., & Singh, H. (2024). Optimised hybrid image encryption scheme using Arnold transform, chaotic map and genetic algorithm for improved security and robustness. AIP Conference Proceedings, 3081(1), 060002. https://doi.org/10.1063/5.0196514