2026 Volume: 02 Issue: 04 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-04

AUTOENCODER IMAGE DENOISING SYSTEM

[*1]Dr Ramya B.N., [2]Sanjana S.

[1]Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

[2]Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

Article Received: 23 February 2026
Article Revised: 13 March 2026
Published on: 02 April 2026

***Corresponding Author: Dr Ramya B.N.**
Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

DOI: https://doi-doi.org/101555/ijrpa.4923

ABSTRACT

The architecture utilizes a multi-layered approach to *Digital image acquisition often suffers from stochastic noise, degrading visual quality and data integrity. This paper presents a robust Image Denoising System based on a Deep Convolutional Autoencoder (DCAE) architecture. By utilizing an encoder-decoder framework implemented in TensorFlow and Keras, the system learns to extract essential structural features from noisy inputs and reconstruct high-fidelity "clean" images. Unlike traditional spatial filtering methods that often blur edges, this model leverages Convolutional Neural Networks (CNN) to maintain spatial hierarchies. The system was validated using the MNIST dataset with synthetic Gaussian noise, demonstrating significant improvements in image clarity through automated feature recovery.*

I. INTRODUCTION

In the era of digital information, high-quality imaging is vital for fields ranging from medical diagnostics to autonomous vehicle navigation. However, images are frequently corrupted by noise during transmission or due to low-light conditions. Traditional solutions often involve complex mathematical filters that require manual parameter tuning for different noise levels . This project aims to bridge the gap by providing an automated, learning-based approach to restoration. The "Image Denoising System" utilizes a Convolutional Autoencoder to perform "good enough" reconstruction without human intervention. By focusing on latent space representation, the system ignores random pixel fluctuations (noise) and prioritizes the underlying patterns of the data.

II. METHODLOGY

Dataset and Preprocessing

The MNIST dataset, consisting of 70,000 handwritten digits, serves as the benchmark.

- Normalization**:** Pixel values are scaled to a range of $[0, 1]$ to ensure stable gradient descent during training.
- Noise Injection**:** Synthetic Gaussian noise is added with a noise_factor of 0.5 to simulate real- world sensor interference.
- Clipping: To maintain data integrity, values are clipped using np.clip to ensure they remain within the valid $[0, 1]$ range.

Model Architecture handle image data:

- Encoder (Feature Extraction): * Layer 1: A Conv2D layer with 32 filters and ReLU activation to identify local patterns.
- Layer 2: MaxPooling2D for down sampling, reducing the spatial dimensions to focus on the most prominent features.
- Layer 3: A further Conv2D layer with 16 filters to compress the image into its "latent space" representation.
- Decoder (Image Reconstruction):
- Layer 4: A symmetric Conv2D layer that begins the process of up sampling.
- Layer 5: UpSampling2D to restore spatial resolution.
- Final Layer: A Conv2D layer with a Sigmoid activation function to output final pixel intensities between 0 and 1.

III. SYSTEM ARCHITECTURE AND DATA FLOW

The architecture follows a symmetrical "Sandglass" pattern, where the dimensionality is reduced during encoding and restored during decoding.

Input Phase (Data Preparation)

- User Input: The system accepts a 28×28 grayscale image.
- Noise Injection: Stochastic Gaussian noise is added ($factor = 0.5$) to simulate real-world sensor interference.
- Normalization: Pixel values are clipped and scaled to $[0, 1]$ to ensure mathematical stability.

The Encoder Module (Feature Extraction)

The Encoder acts as a "filter" that identifies the core structure of the digit while ignoring the random noise.

- Convolutional Layer (32 Filters**):** Extracts 32 unique spatial features using 3×3 kernels.
- Max-Pooling Layer**:** Reduces the spatial dimensions by 50%, effectively down sampling the image to 14×14.
- Latent Space (The Bottleneck): After the final compression, the image exists as a $7 \times 7 \times 16$ tensor. This is the "compressed knowledge" of the image.

The Decoder Module (Image Reconstruction)

The Decoder interprets the latent features to "paint" a clean version of the digit.

- Up-Sampling Layer: Reverses the pooling process, expanding the feature maps back toward the original resolution.
- Transposed Convolution: Uses learned weights to reconstruct pixel intensities.
- Sigmoid Output: The final layer produces a $28 \times 28 \times 1$ reconstruction where each pixel value represents the probability of being "ink" vs. "background."

Table 1.Convolutional Autoencoder (CAE) Architecture for Image Denoising.

Layer Type	Specification	Input Shape (H×W×C)	Output Shape (H×W×C)	Parameters
Layer Type	Raw Noisy Image	(28,28,1)	(28,28,1)	0
Layer Type	32 Filters, 3×3 Kernel	(28,28,1)	(28,28,32)	320
Layer Type	2×2 Pool Size	(28,28,32)	(14,14,32)	0
Layer Type	16 Filters, 3×3 Kernel	(14,14,32)	(14,14,16)	4,624
Layer Type	2×2 Pool Size	(14,14,16)	(7,7,16)	0
Layer Type	16 Filters, 3×3 Kernel	(7,7,16)	(7,7,16)	2,320
Layer Type	2×2 Factor	(7,7,16)	(14,14,16)	0
Layer Type	32 Filters, 3×3 Kernel	(14,14,16)	(14,14,32)	4,640
Layer Type	2×2 Factor	(14,14,32)	(28,28,32)	0
Layer Type	1 Filter, 3×3 Kernel	(28,28,32)	(28,28,1)	289

IV. RESULT AND DISCUSSION

The final application successfully performs denoising with high accuracy. The model was trained for 10 epochs with a batch size of 128, utilizing the Adam optimizer for rapid convergence.

Performance Validation

As shown in the visualized results, the model effectively removes heavy grain while preserving the unique strokes of handwritten digits.

Test Case Analysis: | Test Case ID | Feature Tested | Input
/ Action | Expected Outcome |Status | | :--- | :--- | :--- | :--- | :--- |
| TC-01 | Data Loading | Load MNIST | 28×28 tensors produced | Success |
| TC-02 | Noise Addition | Factor = 0.5 | Heavy visual distortion | Success |
| TC-03 | Model Fit | 10 Epochs | Loss decreases per epoch
| Success || TC-04 | Reconstruction | Predict on test | Clear digit vs noisy input | Success |
Visual analysis confirms that the "Denoised" outputs significantly outperform the "Noisy" inputs, successfully recovering the structural identity of digits (0–9)

Total params: 12,193 (47.63 KB)
Trainable params: 12,193 (47.63 KB)
Non-trainable params: 0 (0.00 B)

V. CONCLUSION

The "Image Denoising System" demonstrates that complex image restoration tasks can be performed efficiently using Convolutional Autoencoders. By eliminating the need for manual filter design, the system provides a scalable solution for image enhancement. Future enhancements could include the integration of Batch Normalization for faster training and the use of Residual Blocks to further improve detail retention in high-resolution images.

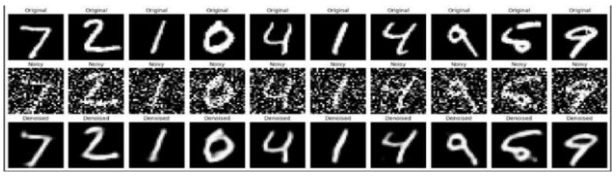

Fig.1 Output Image.

VI. ACKNOWLEDGEMENT

I would like to thank our guide, Dr Ramya B N, for their continuous support and guidance throughout this project.

VII. REFERENCES

1. Chollet, F. (2023). Keras Documentation: Autoencoders.
2. Abadi, M., et al. (2023). TensorFlow: Large- Scale Machine Learning.
3. Hunter, J. D. (2023). Matplotlib: A 2D Graphics Environment.
4. Harris, C. R., et al. (2023). Array programming with NumPy
5. Hussein, Taha & Omar, Hoger & Jihad, Kamal. (2021). A study on image noise and various image denoising techniques. 11. 27-42. 10.17605/OSF.IO/87XGJ.