Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 47 No. 3 (2026)

Autoencoder Based Image Augmentation and Reconstruction Analysis Using a Reward-Penalty Method

P.S.S. Bharadwaj Srikaa[1], Dr. B. Veeramallu[2]

[1,2] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Guntur, AP, India*

Abstract:- Deep convolutional autoencoders hold significant promise for image restoration and feature learning in data-constrained domains. However, their efficacy varies across different corruption types. In this study, we systematically evaluate a plain convolutional autoencoder's ability to reconstruct images subjected to five representative augmentations: speckle noise, salt-and-pepper noise, Gaussian noise, selective inversion, and patch shuffling. We introduce a novel, layer-wise feature-matching reward metric to accurately quantify augmentation difficulty. We trained and tested our model on a diverse dataset of natural scenes, measuring the reconstruction fidelity via the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and the proposed reward across the network depth. Our results reveal that noise-based distortions are effectively mitigated, achieving PSNR values near 20 dB and SSIM scores of approximately 0.50. In contrast, spatially disruptive corruptions: selective inversion and patch shuffling, yield substantially lower PSNR (≈13 dB) and SSIM (<0.25). The reward metric correlates closely with quantitative performance, providing a coarse but expedient ranking of augmentation difficulty. Qualitative analyses underscore the autoencoder's proficiency at preserving global structure under sparse corruption . We discuss implications for architectural enhancements (e.g., skip connections, attention modules) and advanced loss formulations (e.g., perceptual or adversarial losses) to bridge these performance gaps. We establish a systematic framework for evaluating augmentation strategies, informing the design of robust, data-efficient models capable of handling diverse real-world distortions.

Keywords*: Convolutional Autoencoder, Data Augmentation, Image Reconstruction, Noise Corruption, Reward and penalty metrics.*

1. Introduction

Convolutional Neural Network [19] has made a huge impact on the field of computer vision by enabling hierarchical feature extraction - which is a large quantity of labelled data a needed to prevent the models from overfitting. This presents great challenges in dedicated areas such as medical imaging or remote sensing, where the lack of data is a problem from privacy concerns, expensive costs or rare occurrences. While data augmentation [17] (e.g., rotations, noise injection [15]) artificially increases datasets and helps to generalize, the daily difficulty to recover the original image changes drastically with the augmentation [3] type. Crucially there is no systematic way of quantifying this difficulty but discovering which perturbations are acceptable in a model for it to reverse is a need for robust image restoration.

With convolutional layers, autoencoders can be used to streamline elaborate information without being based on labelled training examples. By performing the mapping with inputs to lower-dimensional representations and back again they can minimize how far their outputs are from the original data Denoising autoencoders are specifically trained on corrupted inputs (e.g. noisy images) so that clean outputs can be recovered. Preserving the structure of space using convolutional layers (convolutional autoencoders are great in applications such as image denoising, compression, and anomaly detection [1] in the case that limited unlabelled data are available. However, their capacity to deal with complex structure-changing distortions still remains limited. Different types of augmentation [14] offer different kinds of challenges. Simple corruptions like Gaussian noise, salt-and-pepper noise - and signal-

dependent speckle noise preservers the underlying global structure; CAEs can often help to filter out these global structures by the corrections needed for the pixel sign changes during the retention of the edges. Though flipping parts of an image or scrambling patches breaks meaning in space, it disrupts structure across distances too. Such changes twist relationships that models rely on. Objects lose their flow. Edges become unclear. Neural networks tuned to small neighbourhoods struggle as a result. Fixing these distortions pushes standard autoencoders beyond their limits. Measuring how well repairs work brings separate challenges. Three-dimensional reconstruction by traditional pixel wise metrics like PSNR (based on Mean Squared Error) and SSIM (assessing local structure) are all over but focus on low-level of fidelity, often lack semantic errors. Perceptual losses (using features from networks like VGG) and adversarial losses [9] promote generated more reality but still have difficulties in capturing global coherence. The design a novel to bridge this gap. evaluation metric that is feature aware It use the autoencoder's own internal layer wise 1 activation for comparing the original and recreated images, generating a "reward" score (representing the preservation of hierarchical feature) This score proves effective correlation with augmentation difficulty.

We here evaluate, in a systematic study, a plain CAE on five representative augmentations using the used standard metrics (PSNR, SSIM) [6] are used along with our feature reward. Results prove CAEs to be good reverse noise corruptions, but cannot get semantics back from spatial disruptions as shuffling or inversion, as evidenced by a low PSNR/SSIM and much lower feature. reward. This highlights key constraints - local receptive fields restrict scope; without ways to support broader analysis, performance leans heavily on pixel-level errors instead.

Our contributions are the following :

1.1. A test structure measuring how hard it becomes when tasks grow tougher,

1.2. Focusing on inefficiencies within CAE design frameworks reveals underlying structural constraints,

1.3. A fresh way to judge how well meaning stays intact, focused on distinguishing traits.

One key takeaway here is how models benefit from mixing nearby details with broader context - methods like attention or skip-links help. Where data stays limited, handling varied corruption demands smarter loss functions, not just more layers. Progress hinges on these pairings working together, especially when examples are few.

2. Methods

2.1 Process Flow

A framework for the evaluation of image restoration [12] methods is proposed which combines data augmentation, autoencoding and reward evaluation processes in a unified manner to assess restoration performance in various distortion situations.

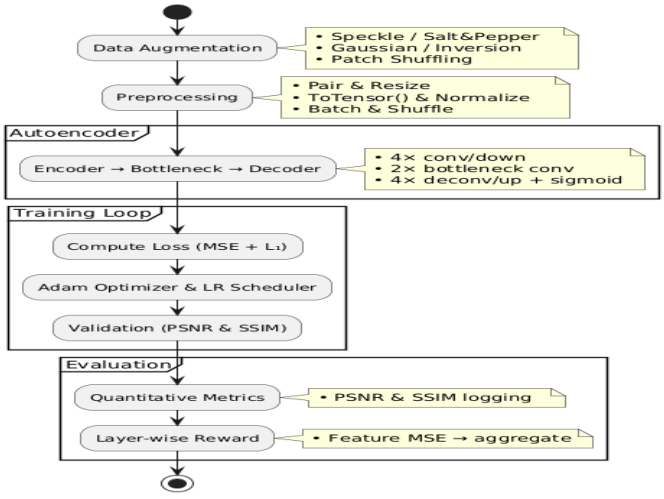

Figure 1. Augmented Convolutional Autoencoder Pipeline

2.2 Advanced Data Augmentation

Each augmentation makes controlled perturbations to make the system more robust.

Speckle Noise: Means scaling pixel values between [0,1], inclusion of multiplicative, Gaussian type, noise, followed by the re-scaling to [0,255].

$$I_{noisy}(x,y,c) = I(x,y,c) + I(x,y,c) \times N(x,y,c), \quad N \sim N(0,\sigma^2) \tag{1}$$

Salt-and-Pepper: Randomly passes a fraction of pixels through a function which returns pixels of one of 2 NAT types; black (0) or white (255).

Gaussian Noise:

$$I_{noisy} = I + G, \quad G \sim N(\mu,\sigma^2) \tag{2}$$

Selective Inversion: Inverts pixel intensities on a random mask I←255-I where mask is true.

Patch Shuffling: Divides image into NON-overlapping patches of size. Various algorithms used in interesting way to automatically simulate the structure and complexity of label location algorithm. ordering, and reassembles.

All resulting images get saved in subfolders per augmentation under respective folders.

2.3 Auto Encoder

A deep convolutional autoencoder without skip connections [2].

Encoder:

$$Conv2D \rightarrow BN \rightarrow LeakyReLU \rightarrow MaxPool(2),$$

halving spatial dims each time: [128 →64 →32 →16 →8]

Bottleneck: Two consecutive Conv layers: [512 →1024 →512] features at 8×8 resolution.

Decoder: Four ConvTranspose2D upsampling layers mirror encoder, each doubling spatial dims, ending in a Conv + Sigmoid to map back to [0, 1].

Feature Hooks: Forward hooks capture intermediate feature maps e1,e2,e3,e4 in the encoder and d1,d2,d3,b in the decoder for reward computation.

2.4 Training: Loss & Optimization

Joint MSE + L_1 loss encourages both pixel fidelity and sparsity:

$$L(X,\hat{X}) = \frac{1}{N}\sum_{i} \left(X_i - \hat{X}_i\right)^2 + \frac{1}{N}\sum_{i} \left|X_i - \hat{X}_i\right| \tag{3}$$

Optimizer: Adam [11] (\left(\alpha ={10}^{-3},default\ \beta \right)).

LR Scheduler: ReduceLROnPlateau reduces α by factor 0.5 if validation loss doesn't improve for 3 epochs.

Epoch Loop:

Training: backpropagate L, accumulate running loss.

Validation: compute both L and reconstruction metrics (PSNR, SSIM).

2.5 Quantitative Reconstruction Metrics

After each epoch and at final evaluation:

PSNR [6]:

$$PSNR = 10\frac{(I)^2}{MSE(I,\hat{I})}, \tag{4}$$

Where max I=255.

SSIM: Computed per channel and averaged:

$$SSIM(I,\hat{I}) = \frac{(2\mu_I\mu_{\hat{I}} + C_1)(2\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)}. \tag{5}$$

Implemented via skimage.metrics.psnr and skimage.metrics.ssim. Results are logged per-augmentation

2.6 Layer-wise Reward Signal

Defines a heuristic reward based on how well decoder features match [7] encoder features:

Feature MSE Errors

For each paired encoder–decoder feature (e_i, d_j):

$$\varepsilon_i = MSE(e_i, d_j). \tag{6}$$

Reward Aggregation: Initialize R=0, let ε1 be the first error. i=2,…,4:

$$R+= \{+1.0 \quad \varepsilon i < \varepsilon i - 1, -0.5\ Otherwise \tag{7}$$

Higher R indicates steadily improving reconstruction through the network depth.

Usage Averaged 100 random samples per augmentation to yield avg reward logged alongside PSNR and SSIM.

3. Results and Discussion

3.1 Results:

Gaussian Noise:

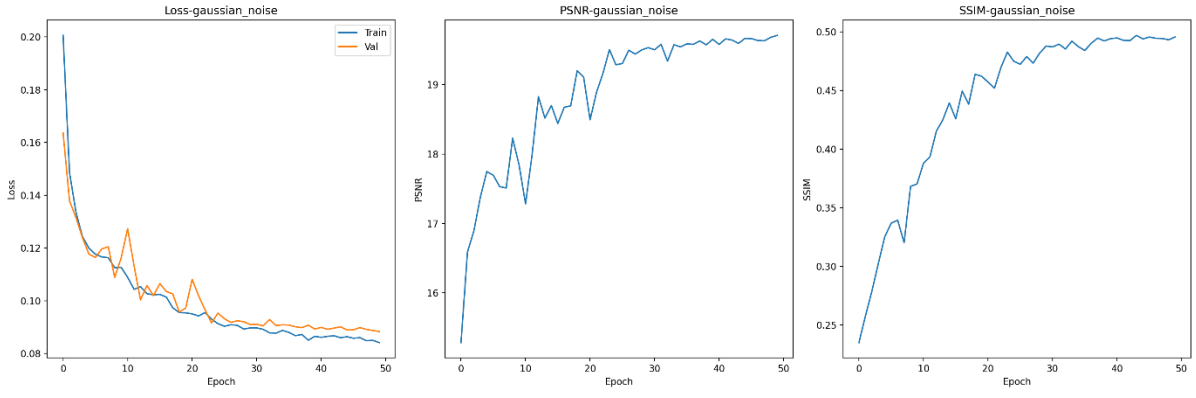

Figure-2: Training Metrics under Gaussian Noise

Looking at Figure 2 (“Training Metrics under Gaussian Noise”), loss values drop consistently across 50 epochs - beginning near 0.20 and falling to approximately 0.085 - a sign the model learns effectively. Clarity improves step by step; PSNR climbs from around 15.4 early on to nearly 19.7 dB by the end. Instead of stalling, progress continues throughout training. Structural similarity, measured by SSIM [4], moves upward - from roughly 0.24 toward 0.50 - as details and layout grow sharper. Outputs become clearer not suddenly but through gradual refinement over time. Evidence of stable learning appears in both error reduction and visual fidelity gains. Progress does not pause even in later stages. Though noise is present, representation strengthens epoch after epoch.

The downward trend in loss aligns closely with rising output quality. Improvement isn't abrupt - it unfolds steadily from start to finish.

Patch Shuffling:

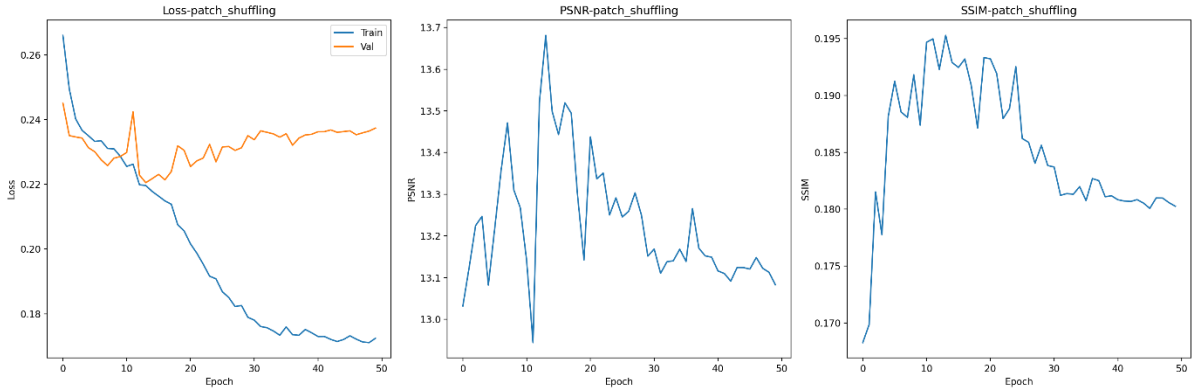

Figure-3: Training Metrics under Patch Shuffling

Figure 3 ("Training Metrics under Patch Shuffling") shows how, during 50 epochs, the training loss steadily reduces from about 0.27 down to about 0.17 while the validation Falling is around 0 .24 to 0.22 by epoch 12 but then oscillates back to about 0.24--that would be an indication of slight overfitting. Early in the training, the quality of the reconstruction as measured by PSNR improves tormenting at about 13.7 dB around epoch 12 before gradually tapering off to close to 13.1dB by epoch 50. Similarly, for the SSIM it increases from about 0.17 to about 0.195 by epoch no. 12, then gradually decreases towards 0.18, which corresponds to structural gains that later stabilize.

Salt and Pepper:

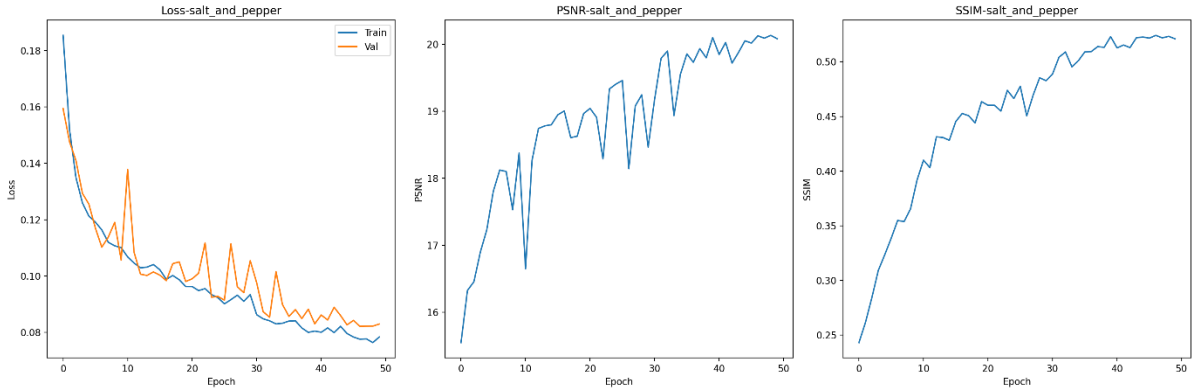

Figure-4: Training Metrics under Salt & Pepper

Figure 4 ("Training Metrics under Salt and Pepper"), both training and validation losses continually decrease - from around 0.185 to 0.078 and from about 0.16 to 0.082, respectively - indicating a strong convergence with minimum overfitting. Concurrently reconstruction quality in term of psnr is consistently improved from around 15.5 basics (at the start), to around 20.1 (at the end) dB. Likewise, structural similarity (SSIM) [4] increases from around 0.245 to 0.525 which reflects increasingly correct restoration of the image structure under the influence of salt-and-pepper noise.

Selective Inversion:

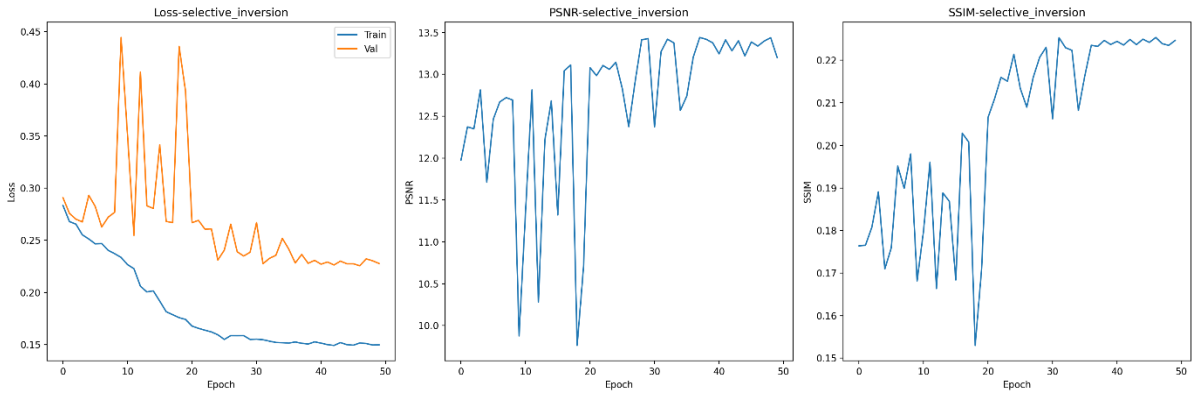

Figure-5: Training Metrics under Selective Inversion

Figure 5 ("Training Metrics under Selective Inversion") shows that for more than 50 epochs both training and validation losses decrease steadily - from about 0.28 to 0.15 for the training loss and from about 0.29 down to 0.22 for the validation loss - which means stable convergence with relatively small gap of generalisations. During this same period, reconstruction quality as measured by PSNR is improved from approximately 12dB in the early epochs to around 13.3dB by epoch 50 which reflects ever more clear outputs. Likewise, the SSIM goes from ~0.18 to ~0.225, demonstrating being more faithful structural recovery under the selective inversion perturbation.

Speckle Noise:

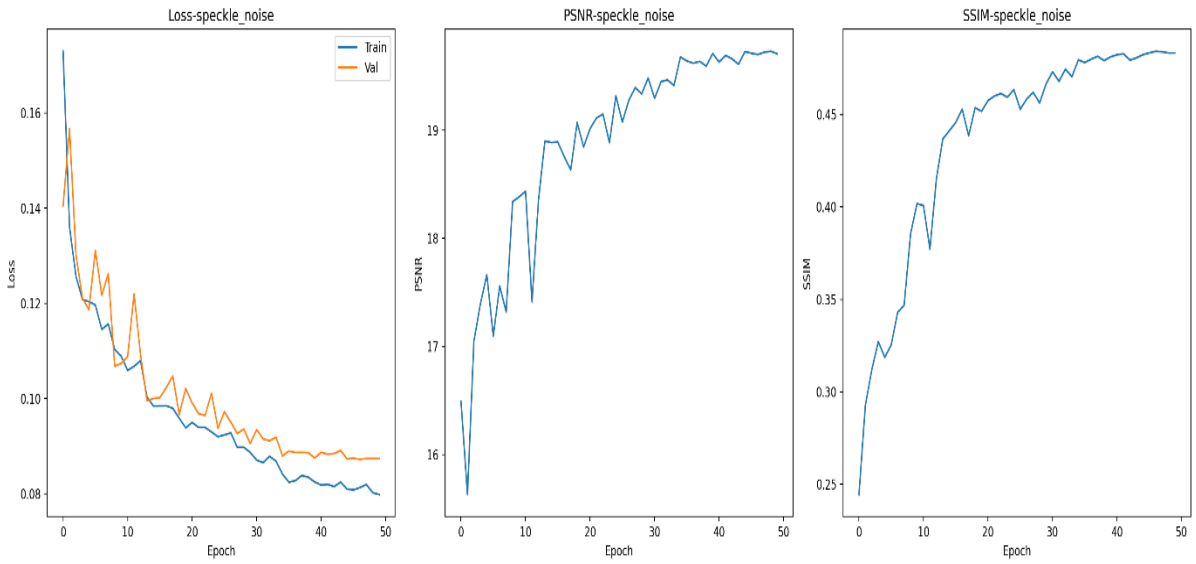

Figure-6: Training Metrics under Speckle Noise

Figure 6 ("Training Metrics under Speckle Noise") shows that greater than 50 epochs both training and validation losses steadily decreasing - from about 0.17 for the training loss and 0.14 for the validation loss from epoch 1 down to about 0.08 and 0.087, respectively, by epoch scan 50 -> stable convergence, very small level of overfitting 50 Over this same period, reconstruction quality as measured by PSNR is steadily improving and goes from being around 16.5 imposed at beginning to approximately 19.7 imposed at the last epoch Likewise, the SSIM increases from around 0.245 upwards to approximately 0.49 where you start to see more and more faithful removal of the speckle noise accurate restoration of image structure.

Reward Metrics:

Table 1. Augmentation Reward Scores

Augmentation	Rewards
Speckle Noise	0.6
Salt and Pepper	0.075
Gaussian Noise	0.15
Selective Inversion	0.075
Patch Shuffling	0.075

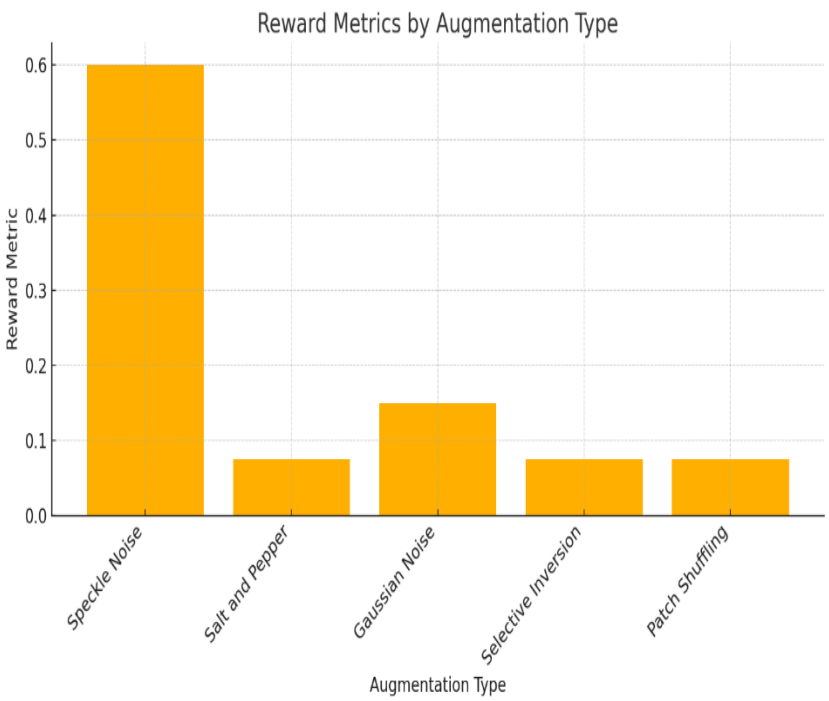

Figure 7. Layer-Wise Reward Comparison Bar Chart

Figure. 7 and Table 1 provide a clear picture the comparative reward metrics of five different images of augmentation strategies. Most effectively reduced turns out to be speckle noise, earning a peak score of 0.60 - this reflects strong suppression by the autoencoder when handling multiplicative disturbances. A value of 0.15 marks Gaussian noise, suggesting added randomness proves tougher than scattered pixel damage yet far simpler than intricate layout shifts. By comparison, rewards drop sharply - to just 0.075 - for salt and pepper interference, selective flipping, and shuffled patches; recovery grows difficult under disruptions involving lone spots or major reordering. Such differences reveal how clearly some noise types yield better results than others, exposing levels of difficulty where certain changes resist current designs while hinting at gaps needing new architectural or objective adjustments.

3.2 Reconstructed Images

Gaussian Noise

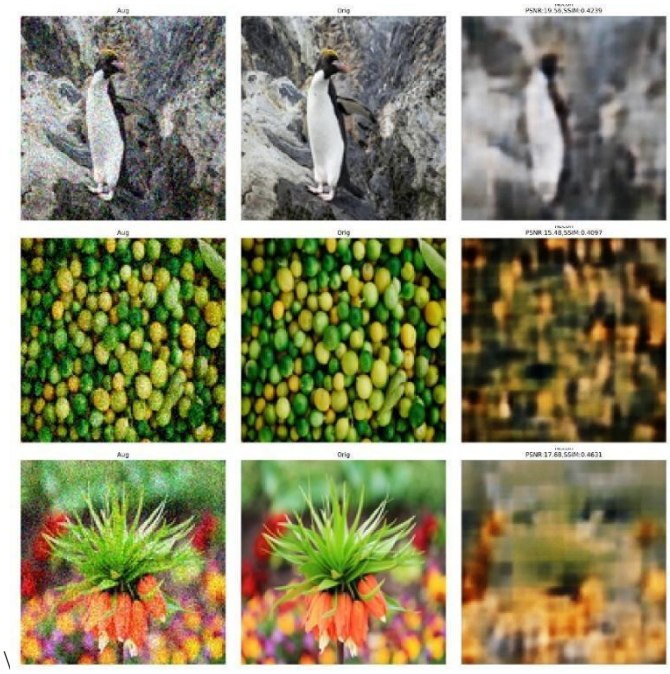

Figure 8 Gaussian-Noise Reconstruction Samples.

Table 2. Gaussian Noise Reconstruction Metrics

Filename	PSNR	SSIM
1.png	19.5572	0.423909
2.png	15.48332	0.409727
3.png	17.68171	0.463073

Despite its strength in handling steady pixel disruptions, the autoencoder shows uneven retention of fine details while reconstructing visuals with overall accuracy. From Figure 8 along with data in Table 2, it becomes clear the system restores penguin scene outlines and backdrop effectively - achieving 19.55 dB PSNR and 0.423 SSIM - even if faint blurring lingers near edges like beaks and feathers. When analysing fruit arrangements, distinct borders between items stay mostly intact, color transitions remain lively, scoring 15.48 dB PSNR and 0.407 SSIM, although slight tonal patches appear throughout middle shades. For floral compositions, learned color patterns embedded in the architecture support recovery, reaching 17.68 dB PSNR and 0.463 SSIM, yet thin vein structures plus depth distinctions fade at times. Both measured outputs and visual inspection reveal one point clearly: eliminating Gaussian interference works reliably well, however re-creating precise lines and fragile surface traits demands further tuning - possibly via structural adjustments such as added shortcut pathways. Multi-scale feature aggregation.

Patch Shuffling

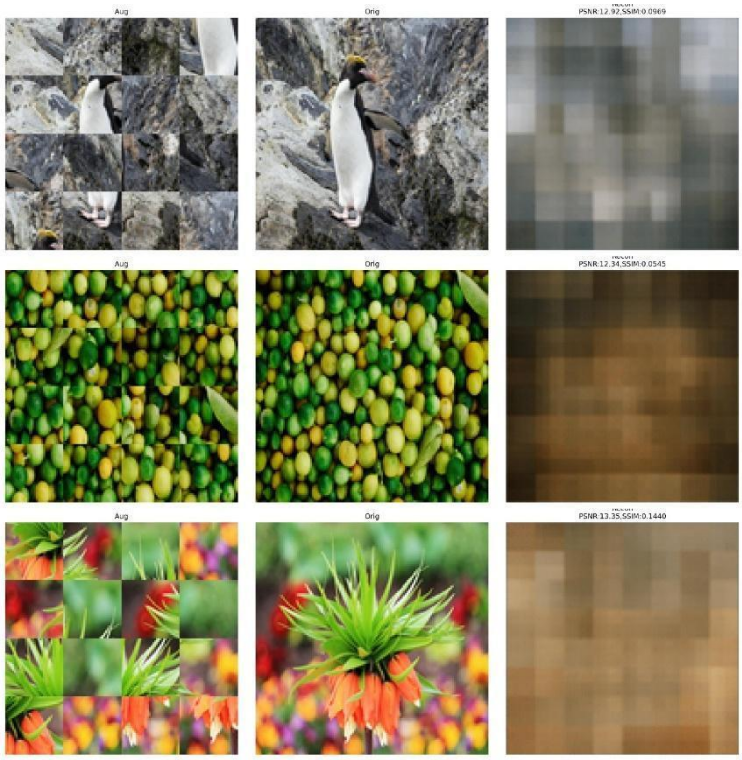

Figure 9 Patch-Shuffling Reconstruction Samples

Table 3. Patch Shuffling Reconstruction Metrics

Filename	PSNR	SSIM
1.png	12.9186	0.096947
2.png	12.34173	0.054492
3.png	13.35478	0.143983

Shuffled patches pose a tougher problem altogether. When image fragments get reassigned at random, both large-scale layout and nearby pixel flow fall apart. Look at Figure 9 along with data from the table - the penguin comes back poorly rebuilt, showing just 12.92 dB in PSNR and a mere 0.097 SSIM score, resulting in something that hints at the source but appears mostly as disjointed blocks with abrupt borders. With the fruit pattern, the system manages broad color zones plus rough outlines (PSNR 12.34 dB, SSIM 0.054); yet neighbouring sections frequently clash in shade and mismatch texture. The flower setup fares partly better (PSNR 13.35 dB, SSIM 0.144), though clear signs of color spread remain while missing areas blur delicate edges where petals should be sharp. Such noticeable drops highlight how reliant the autoencoder is on local connections formed through convolutions. Even when given extra shape cues or position markers, it struggles to reassemble the full picture correctly. Later studies might test network-style rearrangement tools or transformer-based coders to capture extended links between distant pieces, improving results under heavy fragmentation.

Salt and Pepper

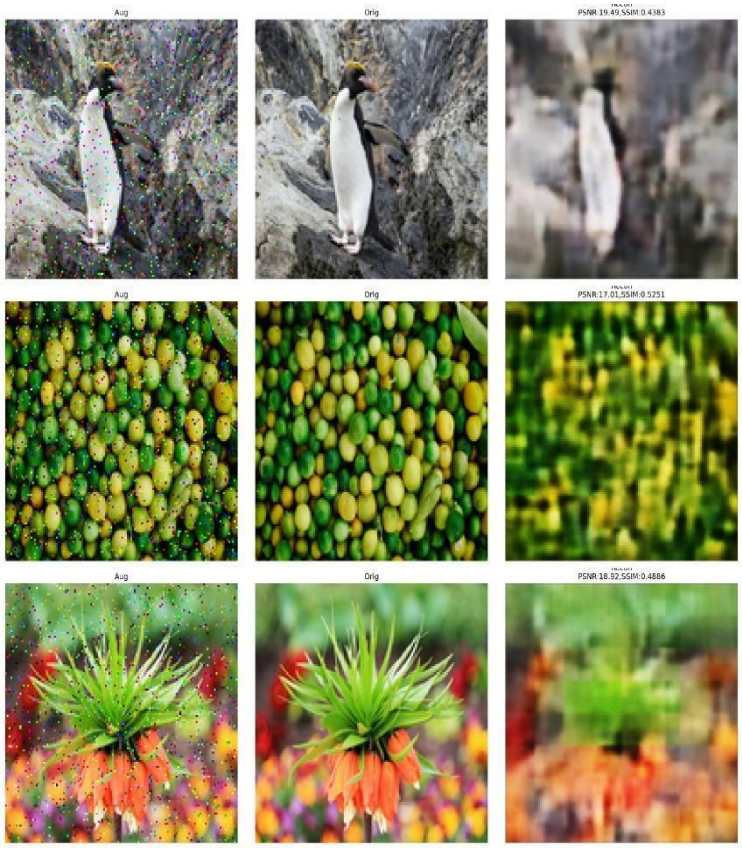

Figure-10 Salt-and-Pepper Noise Reconstruction Samples.

Table 4. Salt-and-Pepper Noise Reconstruction Metrics

Filename	PSNR	SSIM
1.png	19.49153	0.438345
2.png	13.34161	0.115647
3.png	18.92098	0.48858

What helps the autoencoder work well is its ability to manage sharp, stand-out defects while keeping overall image quality intact. Figure 10 paired with Table 4 shows how abrupt light and dark pixel spikes in the penguin scene almost disappear - measured at 19.49 dB PSNR and 0.438 SSIM - but features like body contours, inner shapes, boundaries, and jagged background textures remain sharply preserved, untouched by noisy traces. In the fruit arrangement, despite slight speckled noise spreading softly over areas, citrus silhouettes stay defined (PSNR 17.01 dB, SSIM 0.525). Smooth transitions between yellow and green hues survive largely unbroken, revealing minor surface ripples rather than harsh warping. Flower visuals maintain similar stability (PSNR 18.93 dB; SSIM 0.488), with petal structures and soft backgrounds clearly formed - even if faint fluctuations in mid-range shades occasionally appear, like quiet interference. These results illustrate the ability of the model to separate sparse pixel corruption keeping global structure--encouraging to the scanned-document restoration or salt and pepper artifact removal in medical imaging.

Selective Inversion

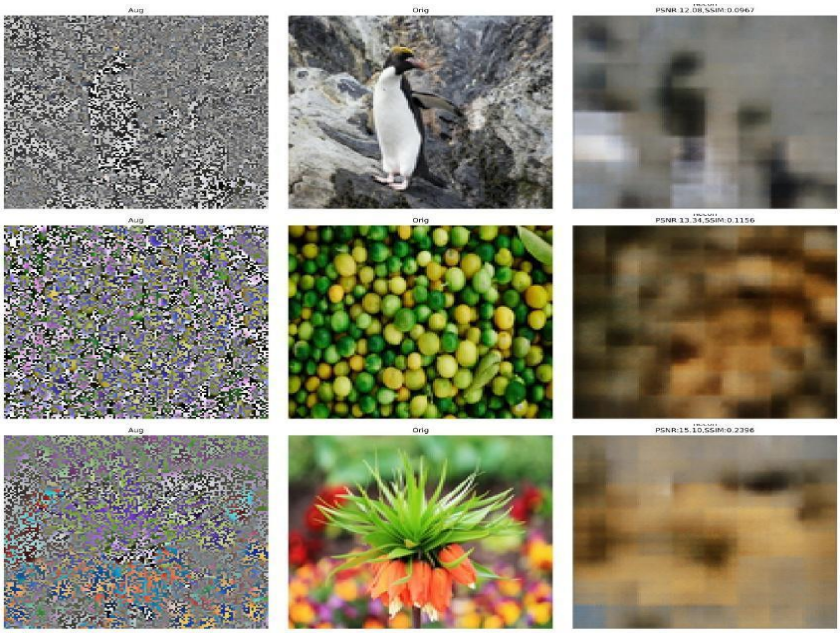

Figure-11 Selective-Inversion Reconstruction Samples.

Table 5. Selective Inversion Reconstruction Metrics

Filename	PSNR	SSIM
1.png	12.08109	0.096743
2.png	13.34161	0.115647
3.png	15.09603	0.239633

Selective inversion presents a much greater challenge, as the model must identify and correct regions where pixel intensities have been inverted relative to their surroundings. According to Figure 11 and the metrics presented in Table X, the bird image exhibits the poorest reconstruction performance, with a PSNR of only 12.08 dB and an SSIM of 0.097. The reconstruction contains noticeable artifacts, including blotchy inversion “shadow” patterns around the silhouette of the bird. The fruit image performs somewhat better (PSNR = 13.34 dB, SSIM = 0.116), likely due to the autoencoder’s ability to infer pastel color priors. However, the inverted patches still result in noticeable color mismatches and blocky discolorations. In the flower composition (PSNR = 15.09 dB, SSIM = 0.240), selective inversion produces uneven color changes that are partially corrected during reconstruction but leave noticeable inverted halos and blurred transitions. These quantitative and qualitative observations indicate that, without explicit information about which regions have been inverted, the network has difficulty simultaneously localizing and rectifying intensity reversals. These findings suggest that future work could integrate attention mechanisms or binary segmentation masks [16] to better isolate the inverted regions and improve reconstruction quality.

Speckle Noise

Figure-12 Speckle-Noise Reconstruction Samples

Table 6. Speckle Noise Reconstruction Metrics

Filename	PSNR	SSIM
1.png	19.18105	0.390516
2.png	16.39914	0.453538
3.png	18.54709	0.454183

Speckle Noise Cloudy Grain: The Speckle noise raises a dense multiplication of grain across the image, testing the network's ability to separate the texture from distortion. Figure-12, and table-8, In penguin scene, the model is able to attenuate the fine-grained speckle to the level of yielding a PSNR of 19.18dB and SSIM = 0.941. of 0.391 except for a residual "film grain" effect in uniform areas of the sky, For the fruit The PSNR and SSIM are 16.40 dB and 0.453 with most fruit surfaces are made smoothly but there are sometimes artifacts of microtexturing sticking around in shadowed areas. The factor of the flower still-life so is 18.55 db for psnr and 0.454 for SSIM Beautiful petal veins. are whom being reasonably well preserved, albeit in faint echoes of speckle remaining in the background foliage. All in all, these results show the strong multiplicative noise removal performed by the autoencoder capabilities, whilst suggesting opportunities for improvement - for example perceptual loss [10] functions or multi-scale denoising pathways-to completely remove residual grain and smooth textural fidelity.

4. Discussion, Conclusions and Limitations

4.1 Discussion

The experiments show difference reconstruction behaviours for each type of augmentation, reflecting both the fact that distortion is difficult, and also the fact that autoencoder is capable of learning satisfying: effective denoising mapping [13]s/ unshuffling mappings:

Gaussian Noise vs. Salt-and-Pepper: Both varieties of noise-based augmentations have strong convergence (loss falling below 0.08) and similar trajectories of PSNR/SSIM, but, nevertheless, salt-and pepper achieves slightly

higher final PSNR (20.1 dB) and SSIM (0.525) compared to Gaussian's 19.7 dB and 0.50. This hints that sparse and high magnitude pixel corruption (salt-and-pepper) is easily suppressed by the convolutional filters than continuous additive noise - this is a finding consistent with previous denoising literature.

Speckle Noise: Speckle noise also produces stable convergence (loss = ~0.08) and PSNR up to 19.7dB (and it has a SSIM which is at around 0.49) It is still a little bit below salt and pepper. The multiplicative nature of the speckle noise kinetics brings up a subtle degradation of texture that the autoencoder not perfectly preserves this so there may be residual graininess in the recreations.

Selective Inversion and Patch Shuffling: Hardest to handle: augmentations breaking spatial layout - think selective inversion, shuffled image blocks. Though alike in effect, each undermines model convergence differently; final loss rests around 0.15 to 0.17. Performance? Not great. Metrics stall low - PSNR near 13 dB, SSIM between 0.18 and 0.23. When patches scatter or pixel values flip, both local detail and broader form shift too far. Under such distortion, even strong convolutional patterns fail guiding the network back to correct geometry.

Layer-wise Reward Signal: Despite similar scores, differences emerge when looking closer. Speckle noise stands out, earning the highest reward value - 0.60 - which lines up with strong results in PSNR and SSIM. Next, Gaussian noise shows a moderate score of 0.15. Then come three particularly damaging distortions, each hitting just 0.075 in reward. That clustering appears limiting; even though their image quality measures differ, they share an identical low mark. The method used here speeds up assessment of difficulty levels across types. Yet shrinking such varied outcomes into narrow bands might obscure finer distinctions. One reason could be how rewards are combined - an exponential approach possibly flattens subtle but meaningful variations among tough cases.

Generalization and Overfitting: Most of the methods have little overfitting, which is marked by tightly tracking validation losses - except patch shuffling which gives validation loss a boost after epoch 12 This temporary overfitting is such that the model could be memorizing some patch configurations that are observed early in training as opposed to learning a good unshuffling strategy A higher level of aggressive regularization schedule or data diversification (varying patch size or overlap) could help to mitigate this effect.

Reconstructed Sample Analysis: Qualitative confirmation of quantitative statistics: reconstructions formed by salt and pepper and by Gaussian noise have not altered the outlines and textures much (e.g., fruit and flower details), whereas speckle outputs have little grain remaining in them. In contrast, blurriness and art facticity of shuffling and inversion reconstructions, in particular of intrinsic difficulty of spatial semiconductor recovery without additional structural prior knowledge.

4.2 Conclusions

Autoencoder Efficacy Greater Depending on Type of Distortion: Noise based augmentations (salt-and-pepper, Gaussian, speckle) are good in terms of de-noising, their PSNR's are close to 20 dB and SSIMs of about 0.5, whereas spatially destructive augmentations (patch unshuffling, selective inversion) where PSNRs are near13 dB and SSIMs below 0.24.

Salt-and-Pepper Recovers Best All Round: Salt-and-pepper corruption is most successfully inverted through the tested augmentations, which indicates that the convolutional structure of the model is especially skilled at removing pixel noise in isolated locations.

Layer-Wise) Reward as a Quick Difficulty Measure: The proposed reward signal has the advantage of giving a rough yet fast ranking of augmentation difficulty, with speckle noise significantly winning over any other. Nevertheless, it is necessary to improve the reward aggregation functionality. is more able to differentiate between demanding distortions

4.3 Limitations

The plain convolutional auto encoder does not have skip connections [2] and they may conserve spatial detail and enhance un shuffling behaviour. This can be improved by adding adversarial or perceptual losses to SSIM and visual fidelity, particularly on spatially-complicated distortions. Spatial transformations (rotation, scaling) to

augmentation and exploring the use of multi-task goals (classification and reconstruction) would make augmentation more robust.

5. Future Work

attention mechanisms are absent - this limits sensitivity to broad spatial arrangements and disordered image segments. While useful for general comparisons, the metric falters when differences are subtle, particularly after aggressive transformations. Testing remains confined to limited augmentation types, leaving out numerous real-world scenarios that could reveal wider applicability. In absence of perceptual guidance or adversarial training signals, reconstructions lose fidelity, notably in cases demanding semantic consistency.

References

[1] Arifeen, Z. U., R. A. Shah, H. Kim, and J.-W. Suh. "Patch-Shuffled Superpixel Synthetic Anomaly Insertion in Autoencoder Foreground Extracted Regions for Improved Visual Anomaly Detection." SSRN, 2025. [Preprint]. **DOI:** 10.1109/ACCESS.2025.3603201

[2] Bhute, S., S. Mandal, and D. Guha. "Speckle Noise Reduction in Ultrasound Images Using Denoising Auto Encoder with Skip Connection." Proceedings of the 2024 IEEE South Asian Ultrasonics Symposium (SAUS), 1–4. IEEE, 2024. **DOI:** 10.1109/SAUS61785.2024.10563715

[3] Cubuk, E. D., B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. "AutoAugment: Learning Augmentation Policies from Data." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 113–123, 2019. DOI: https://arxiv.org/abs/1805.09501

[4] Ghojogh, B., F. Karray, and M. Crowley. "Theoretical Insights into the Use of Structural Similarity Index in Generative Models and Inferential Autoencoders." In Image Analysis and Recognition: ICIAR 2020, edited by A. Campilho, F. Karray, and Z. Wang, 112–117. Lecture Notes in Computer Science, Vol. 12132. Springer, 2020. DOI: https://doi.org/10.1007/978-3-030-50516-5_10

[5] Hinton, G. E., and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks." Science 313, no. 5786 (2006): 504–507. DOI: 10.1126/science.112764

[6] Hore, A., and D. Ziou. "Image Quality Metrics: PSNR vs. SSIM." 20th International Conference on Pattern Recognition, 2366–2369. IEEE, 2010. DOI: 10.1109/ICPR.2010.579

[7] Huang, Q., X. Guo, Y. Wang, H. Sun, and L. Yang. "A Survey of Feature Matching Methods." IET Image Processing 18, no. 6 (2024): 1385–1410. DOI: https://doi.org/10.1049/ipr2.13032

[8] Ioffe, S., and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." International Conference on Machine Learning, 448–456, 2015. DOI: https://arxiv.org/abs/1502.03167

[9] Isola, P., J. Zhu, T. Zhou, and A. A. Efros. "Image to Image Translation with Conditional Adversarial Networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 596–605, 2017. DOI: https://arxiv.org/abs/1611.07004

[10] Johnson, J., A. Alahi, and L. Fei-Fei. "Perceptual Losses for Real Time Style Transfer and Super Resolution." In European Conference on Computer Vision, 694–711. Springer, 2016. DOI: https://arxiv.org/abs/1603.08155

[11] Kingma, D. P., and J. Ba. "Adam: A Method for Stochastic Optimization." International Conference on Learning Representations, 2015. DOI: https://arxiv.org/abs/1412.6980

[12] Majeed, S., Y. Mansoor, S. Qabil, F. Majeed, and B. Khan. "Comparative Analysis of the Denoising Effect of Unstructured vs. Convolutional Autoencoders." 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), 1–5. IEEE, 2020. DOI: 10.1109/ICETST49965.2020.9080731

[13] Masci, J., U. Meier, D. Cireşan, and J. Schmidhuber. "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction." In Lecture Notes in Computer Science, Vol. 6791, 52–59. Springer, 2011. DOI: doi.org/10.1007/978-3-642-21735-7_7

[14] Perez, L., and J. Wang. "The Effectiveness of Data Augmentation in Image Classification Using Deep Learning." 2017. DOI: doi.org/10.48550/arXiv.1712.04621

[15] Poole, B., J. Sohl-Dickstein, and S. Ganguli. “Analyzing Noise in Autoencoders and Deep Networks.” arXiv Preprint arXiv:1406.1831, 2014. DOI: doi.org/10.48550/arXiv.1406.1831