International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.15, No.1, February 2026

ANALYSIS OF THE IMPACT OF EIGHT-BIT REDUCED PRECISION ON IMAGE RECONSTRUCTION FIDELITY

Ricardo Francisco Martínez-González [1] and Abelardo Rodríguez-León [2]

[1] Department of Electrics and Electronics, Tecnológico Nacional de México – IT Veracruz, Veracruz, México
[2] Department of Mechanics, Tecnológico Nacional de México – IT Veracruz, Veracruz, México

ABSTRACT

This study evaluates the impact of transitioning from FP64 to 8-bit floating-point (FP8 E4M3) for image reconstruction using autoencoders. To address energy and memory constraints in edge computing, we apply Post-Training Quantization with dynamic scaling (range 448). Although quantization increases Mean Squared Error slightly, perceptual quality remains acceptable. Memory use per parameter drops by 87.5%, drastically reducing computational load while preserving utility. The findings confirm that low-precision formats enable efficient AI deployment on resource-limited devices, allowing embedded vision systems to operate with significantly lower energy overhead.

KEYWORDS

Post-Training Quantization, FP8 E4M3 Format, Reconstruction Fidelity

1. INTRODUCTION

Efficient execution of computer vision algorithms on resource constrained platforms—such as edge devices and smart sensors—has intensified the shift from high precision numeric formats toward reduced arithmetic architectures [1]. For years, the IEEE 754 double precision standard (64 bit) has dominated scientific computing owing to its wide dynamic range and negligible rounding error. Yet in contemporary deep learning, this abundance of numerical resolution often becomes a source of inefficiency. Transporting high precision data between memory and processors can consume up to 90 % of the total energy during inference, rendering bit depth reduction an imperative not only for architecture but also for energy and thermal management [2].

Moving to narrower formats is not a simple truncation exercise. While fixed point (INT8) has been widely adopted for its simplicity in arithmetic logic unit (ALU) design, the eight bit floating point format (FP8) offers a more capable alternative for representing neural network parameters [3]. Its strength lies in a split binary structure of sign, exponent, and mantissa. Unlike the linear spacing of fixed point, floating point provides logarithmic density, which is crucial in modern networks where weights and biases frequently follow normal or Laplace distributions clustered near zero [4]. Employing a variant such as E4M3 (4 exponent bits, 3 mantissa bits) helps capture the subtle stochasticity of these parameters while preserving consistent relative precision across multiple orders of magnitude.

DOI : 10.5121/ijscai.2026.15101

The consequences of precision loss, however, depend heavily on the task. In classification, where the goal is to map high dimensional inputs to discrete labels, networks often exhibit intrinsic robustness to quantization; errors in intermediate layers can be tolerated provided the correct class retains the highest probability [5]. Image reconstruction, by contrast, requires precise regression per pixel. Here, signal fidelity is paramount. Excessive bit reduction manifests as "quantization noise," whereby weight discretization impedes the representation of smooth gradients. Visually, this can produce banding or posterization, fracturing continuous luminance transitions into discrete steps and thereby degrading structural integrity [6].

From an information theory standpoint, lowering precision from 64 to 8 bits implies a theoretical data compression of 87.5 %. The scientific question is whether such aggressive compression can be offset through dynamic scaling and bias adjustment. This work implements an autoencoder tasked with encoding and reconstructing visual information using a decoder optimized entirely in FP8. Through this experiment, we quantify the MSE and qualitatively assess texture and luminance preservation, determining whether reduced floating point precision can sustain the continuity required for human perception in high efficiency systems [7].

2. Development

Our methodology rests on the framework of Inference Optimization via Post Training Quantization (PTQ). This approach exploits the parametric redundancy present in neural networks after they have converged in high precision, allowing a direct transformation of weight tensors WFP64→WFP8 [8]. Unlike Quantization Aware Training, which demands extra computation and large datasets, PTQ seeks a streamlined conversion to lower precision. The central aim is to identify the threshold of acceptable information loss where the energy efficiency of eight bit arithmetic does not undermine the structural integrity of the reconstructed image. This stage is critical for embedded system design, where memory management units (MMUs) and data bus bandwidth impose strict limits on information flow [9].

2.1. Foundations and Range Calculation of the FP8 (E4M3) Format

Before software implementation, we must establish the mathematical boundaries of the FP8 E4M3 format. This eight bit layout allocates one bit to the sign (s), four to the exponent (e), and three to the mantissa (m). The format prioritizes mantissa resolution over dynamic range, making it well suited for visual signal reconstruction [10].

Representable values follow the bias 7 formula

$$V = (-1)^{8} \times 2^{e-7} \times (1 + m \times 2^{-3}) \qquad \text{Eq. 1}$$

The operational limits derived for this study are:
Maximum value (Vmax): Using the maximum normalized exponent (binary 1110, decimal 14) and a full mantissa (binary 111, decimal 0.875):

$$V_{max} = 2^{14-7} \times (1 + 0.875) = 128 \times 1.875 = 240 \qquad \text{Eq. 2}$$

Minimum precision: The smallest step between adjacent values is governed by the least-significant mantissa bit (2−3=0.125).

2.2. Visual Stimulus Generation and Normalization

With arithmetic limits defined, the first coding step creates an analytical test pattern. We generated a synthetic 16 × 16 pixel diagonal gradient. The purpose is to define a continuous intensity function f(x,y) that is normalized to the interval [0,1] in double precision (FP64), ensuring the signal enters the network with maximal fidelity [11].

Code 1. Creation of a test signal with high spatial variance

```
rows, cols = np.indexes((16, 16))
image_np = ((rows + cols) * (255 / 30)).astype(np.uint8)
original_input = torch.from_numpy(image_np).view(1, -1).double() / 255.0
```

2.3. Scaling Algorithm and FP8 Arithmetic Simulation

This block translates the theoretical limits from Section 2.1 into computational operations. The goal is to map 64 bit tensors into the E4M3 "funnel." A scale factor S projects the tensor's maximum value toward the 448 limit. Applying torch.clamp simulates hardware saturation, revealing how bit loss affects gradient smoothness [12].

Code 2. Definition of the dynamic scaling function

```
def quantize_to_fp8_simulated(tensor):
    max_val = tensor.abs().max()
    scale = 448 / (max_val + 1e-8) # Dynamics adjustment FP8
    q_tensor = tensor * scale
    q_tensor = torch.clamp(torch.round(q_tensor), -448, 448)
    return q_tensor / scale
```

2.4. Autoencoder Architecture and Latent Reduction

The architecture is a symmetric "identity mirror" with two dense layers of 256 neurons each. Its purpose is to evaluate the network's capacity to compress the image into a latent space and recover it. The Sigmoid activation maps outputs to [0,1][0,1], emulating the luminance response of a physical sensor and forcing quantized weights to maintain fine precision to avoid visible banding [8].

Code 3. Definition of the Autoencoder System architecture

```
class AutoencoderSystem(nn.Module):
    def __init__(self):
        super().__init__()
        self.net1 = nn.Linear(256, 256) # Coder
        self.net2 = nn.Linear(256, 256) # Decoder
        self.activation = nn.Sigmoid()
```

2.5. Optimization and Convergence

A high fidelity baseline is established by training the model in FP64 for 1 500 epochs. This guarantees that any degradation observed later stems from bit reduction, not inadequate training. The target is an MSE near zero before quantization [9].

Code 4. High-precision training process (FP64)

```
optimizer = torch.optim.Adam(model.parameters(), lr=0.005)
criterion = nn.MSELoss()
```

```
for epoch in range(1500):
  optimizer.zero_grad()
  persistent_noise, recovered = model(original_input)
  loss = criterion(recovered, original_input)
  loss.backward()
    optimizer.step()
```

2.6. Extraction and Evaluation of Net 4 (FP8 with Adjusted Bias)

The final step is post training quantization of the Net 2 parameters (weights and biases). The aim is to gauge the network's resilience when its parameters become discrete. This involves creating Net 4_FP8, which uses the already "degraded" values to perform the final reconstruction, enabling a comparative MSE analysis against the original 64 bit version [10].

Code 5. Implementation of Net 4 with quantized parameters

```
with torch.no_grad():
  pesos_fp8 = quantize_to_fp8_simulated(modelo.red2.weight.detach())
  bias_fp8 = quantize_to_fp8_simulated(modelo.red2.bias.detach())

  red4 = Red4_FP8(pesos_fp8, bias_fp8).double()
  recuperada_fp8 = red4(ruido_f)
```

3. Results

System performance was assessed by comparing the synthesis capability of the original network against the reduced precision network, linking qualitative visual evidence with quantitative metrics. The findings illustrate how precision reduction influences both signal morphology and system efficiency.

3.1. Comparative Visual Analysis and Signal Process

Results are displayed in a four stage inspection matrix (Figure 1), tracing the signal's evolution from origin to final recovery. This approach helps detect spatial artifacts that global metrics might miss in digital image processing [11]. The first quadrant shows the Net 1 input: the original reference pattern, a perfect linear gradient with continuous, noise free intensity transitions. The Net 1 output presents the latent signal—the abstract representation that subsequent networks must interpret to reconstruct the image.

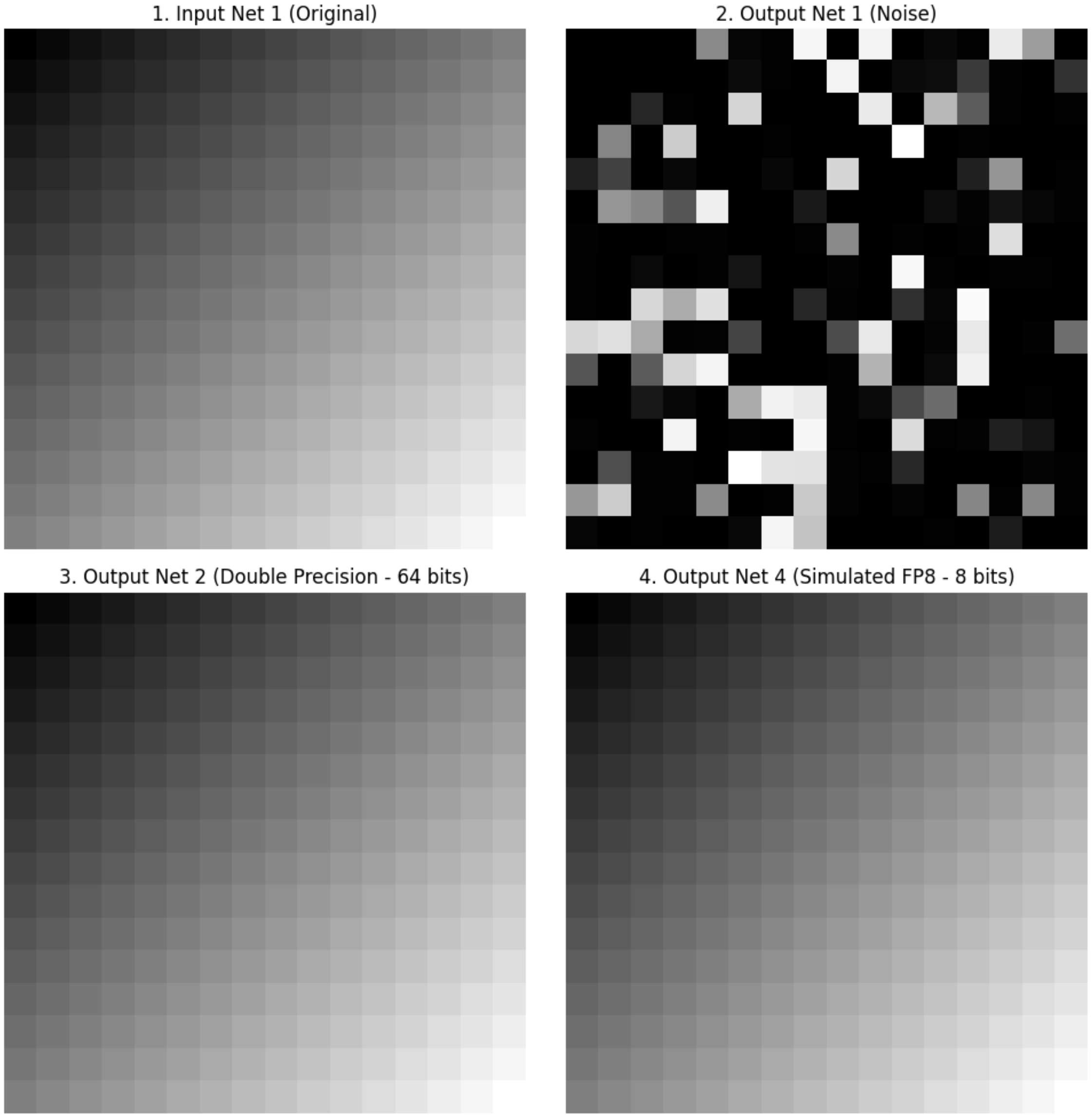

Figure 1. Multi-stage comparison of the reconstruction process: from the original stimulus and its latent encoding to the final synthesis in high precision (FP64) and reduced precision (FP8).

Comparing the two recovery methods reveals clear technical differences. The output from the high precision network (FP64) is visually indistinguishable from the original, confirming that 64 bit arithmetic can map the latent signal back to pixel space with near absolute fidelity. The signal recovered by the simulated FP8 network preserves the overall gradient structure but introduces subtle irregularities in tonal smoothness. These irregularities arise directly from limiting the mantissa to three bits, which restricts the number of intermediate gray levels a neuron can compute before activation [4].

3.2. Quantitative Analysis of Mean Squared Error (MSE)

To substantiate the visual observations, we computed the Mean Squared Error (MSE). This metric measures the average squared discrepancy between the original image (I) and the recovered image (Y) on a pixel by pixel basis, converting qualitative perception into rigorous statistics.

Calculation Methodology. Original intensity levels were normalized to the floating point interval [0,1]. The MSE is defined as

$$MSE=\frac{1}{n}\sum_{i=1}^{n}(I_i-Y_i)^2$$

Eq. 3

where n=256 (the total number of pixels in the 16 × 16 matrix). This procedure quantifies brightness deviation of each reconstructed unit from its ideal value, indicating how closely the network approaches perfect reconstruction [12].

3.3. Performance Comparison.

The experimental data reflect a balanced trade off between arithmetic precision and computational savings. Table 1 summarizes the numerical outcomes.

Table 1. Performance and efficiency comparison in image reconstruction: Double-Precision (FP64) vs. Eight-Bit Floating-Point (FP8).

Performance metric	Net 2 (FP64)	Net 4 (FP8)	Difference / Impact
Average MSE	0.0000008431	0.0000492105	+0.00004836
Memory load	64 bits/param.	8 bits/param.	87.5% saving
Visual fidelity	Absolute	High (noisy)	Mantissa loss

Although the eight bit network's error is orders of magnitude larger than the original, the absolute MSE of Net 4 remains extremely low. This confirms that dynamic scaling to the 448 range successfully placed weights in the highest resolution zone of the E4M3 format, mitigating quantization error. The outcome is a system that retains gradient structural integrity while drastically shrinking memory footprint—a crucial feature for deploying neural networks on resource limited hardware [13].

4. CONCLUSIONS

This analysis demonstrates that the move toward reduced precision formats, specifically the FP8 E4M3 standard, is not merely a technical optimization but a viable strategy for preserving complex visual signals. The study shows that quantization's impact on image fidelity remains remarkably small when a suitable dynamic scaling algorithm is applied, allowing gradient structural integrity to be maintained without distortions that would hinder interpretation or downstream use [6].

A key finding is the substantial resource saving at the coefficient level. By shrinking each parameter's representation from 64 to 8 bits, we achieve an 87.5 % compression in the model's memory footprint. This drastic reduction in neural network coefficients is decisive for implementing computer vision algorithms on resource constrained devices—ultra low power microcontrollers, smart sensors, and Edge AI hardware. In such environments, where storage and memory bandwidth are primary bottlenecks, operating with eight bit tensors enables more capable models without raising hardware costs or energy consumption [14].

Finally, both quantitative and qualitative results confirm that recovered image quality is not severely compromised. The modest increase in MSE is a marginal cost compared to the gains in operational efficiency. The resilience exhibited by the autoencoder under eight bit arithmetic suggests that the future of real time image processing depends not on numerical brute force, but on specialized arithmetic formats that, like FP8, adeptly balance needed precision with the resource austerity demanded by today's technology industry [8].

REFERENCES

[1] Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., & Soudry, D. (2021). Accurate post training quantization with second order information. Proceedings of the 38th International Conference on Machine Learning (PMLR 139), 4487–4496.

[2] Wang, E., Davis, J. J., Cheung, P. Y. K., & Luk, W. (2022). Energy efficient deep learning inference with fixed point arithmetic. IEEE Micro, 42(3), 45–53. https://doi.org/10.1109/MM.2022.3161916

[3] Kuzmin, A., van Baalen, M., Ren, Y., Nagel, M., Peters, J., & Blankevoort, T. (2023). FP8 versus INT8 for efficient deep learning inference. Proceedings of the 40th International Conference on Machine Learning.

[4] Micikevicius, P., Stosic, B., Burgess, N., Cornea, M., Dubey, P., Grisenthwaite, R., … & Wu, H. (2022). FP8 formats for deep learning. arXiv preprint arXiv:2209.05433.

[5] Choi, J., Wang, Z., Venkataramani, S., Chuang, P. I. J., Srinivasan, V., & Gopalakrishnan, K. (2018). PACT: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085.

[6] Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M., & Blankevoort, T. (2021). A white paper on neural network quantization. arXiv preprint arXiv:2106.08295.

[7] Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. International Conference on Learning Representations (ICLR).

[8] Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv preprint arXiv:1806.08342.

[9] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations (ICLR).

[10] Noune, B., Jones, P., Justus, D., Masters, D., & Luschi, C. (2022). 8-bit floating point formats for deep learning. Proceedings of the 2022 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS).

[11] IEEE Computer Society. (2019). IEEE Standard for Floating Point Arithmetic (IEEE Std 754 2019).

[12] Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing (4th ed.). Pearson.

[13] Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2020). Efficient processing of deep neural networks. Morgan & Claypool Publishers.

[14] Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference. arXiv preprint arXiv:2103.13630.