Convolutional Autoencoder Compression for LoRa Visual Transmission on Embedded Systems

João Mendes[1,2(✉)], Inês Sena[1,2], José Lima[1,2], and Ana I. Pereira[1,2]

[1] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
{joao.cmendes,jllima,apereira}@ipb.pt

[2] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
ines.sena@ipb.pt

Abstract. This work proposes and evaluates a lightweight convolutional autoencoder (CAE) designed for image compression in Internet of Things (IoT) nodes subject to severe bandwidth constraints. The system follows an encode–transmit–decode paradigm: a compact encoder, intended for deployment on an ESP32 microcontroller, compresses a 96×96 pixel grayscale image into a low-dimensional latent vector; this vector is then transmitted over LoRa as a single packet or multiple fragments; and a decoder at the receiver reconstructs the image from the received representation. Seven latent dimensions ($d \in \{8, 16, 32, 64, 128, 256, 512\}$) are systematically evaluated and compared against a Principal Component Analysis (PCA) baseline using Peak Signal-to-Noise Ratio (PSNR, measured in decibels) and the Structural Similarity Index Measure (SSIM) on the MNIST dataset. Results show that the CAE substantially outperforms PCA in all single-packet LoRa configurations ($d \leq 64$), achieving reconstructions that surpass commonly accepted quality thresholds while reducing the transmitted payload to a single 64-byte LoRa packet, a 144:1 reduction relative to the raw image. These findings demonstrate that non-linear learned representations are substantially more efficient than linear projections under the low-dimensional regime imposed by LoRa bandwidth constraints.

Keywords: Wireless Visual Sensor Networks · LoRa · Convolutional Autoencoder · Learned Image Compression · TinyML · Internet of Things

1 Introduction

The use of Wireless Sensor Networks (WSN) in the context of the Internet of Things (IoT) has intensified the requirement for efficient transmission of visual

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16759, pp. 210–223, 2027.
https://doi.org/10.1007/978-3-032-30518-3_14

information from embedded nodes with severely constrained computational and energy resources [1,2]. In sectors such as precision agriculture, environmental monitoring, and remote infrastructure management, the capacity to capture and communicate image data from low-cost devices is a growing, yet unresolved challenge [3,4].

Unlike scalar sensor data, images carry far more information per sample, and that comes at a cost. Wireless Visual Sensor Networks (WVSN) must deal with data volumes and structural complexity that standard WSN architectures were simply not built for, making the design of acquisition, processing, and transmission pipelines a genuinely difficult engineering problem [5].

In this context, LoRa (Long Range) technology is among the most promising wireless communication solutions for a wide range IoT applications [6]. LoRa modulation, based on Chirp Spread Spectrum (CSS), enables communication ranges up to 10 km in rural environments while maintaining "listening" mode energy consumption on the order of microwatts, making it especially suitable for battery- or solar-powered nodes [7]. However, the actual application bandwidth is typically below 2 kbps, and the maximum payload per packet is limited to approximately 115 bytes under common configurations (SF9, 125 kHz, 868 MHz band), resulting in severe restrictions on data transmission capacity [8,9].

Under these conditions, direct transmission of uncompressed images is entirely impossible. For example, a grayscale image of size 96×96 pixels requires 9,216 bytes, equivalent to more than 80 LoRa packets and a transmission time exceeding 40 s, without accounting for the duty cycle restrictions imposed by ETSI EN 300 220-2. Therefore, image compression is required for any system that transmits visual data over LoRa.

The main problem is that classic techniques of compression such as JPEG, which are widely available even on low-end microcontrollers [10], are designed for moderate compression with minimal perceptual loss, and its performance degrades significantly at very low bitrates, the exact regime in which LoRa systems operate. JPEG compression at payloads of 64 or 128 bytes introduces severe blocking artifacts and structural detail loss, compromising the use of the reconstructed image [11].

One alternative to these methods is the convolutional autoencoder (CAE). These models are trained to learn compact, informative representations of images [12,13]. CAEs learn nonlinear transformations of the input data, preserving semantically relevant features while using significantly fewer bytes [14].In the encoder, executed on the embedded device, a compression of the image into a low-dimensional latent vector. This is transmitted via LoRa and the decoder component, executed on the receiver server, reconstructs the image from the received representation.

All the feasibility of this method raises several research questions:

(i) What is the minimum latent vector size required to achieve acceptable visual quality, defined here as Peak Signal-to-Noise Ratio (PSNR, a logarithmic measure of reconstruction fidelity expressed in decibels) $\geq$ 25 dB and Struc-

tural Similarity Index Measure (SSIM, a perceptual metric ranging from 0 to 1) ≥ 0.90?

(ii) How does the convolutional autoencoder compare with Principal Component Analysis (PCA), a classical linear compression baseline [15], in terms of PSNR and SSIM for the same payload?

(iii) What latent dimension best balances reconstruction quality and transmission efficiency given the single-packet LoRa boundary ($d \leq 112$ bytes of useful payload)?

On top of this, executing the encoder on a low-cost microcontroller, specifically the ESP32-S3, featuring an Xtensa LX7 processor at 240 MHz and support for neural inference via TensorFlow Lite Micro introduces extra constraints on model size and complexity, requiring the evaluation of integer quantization (INT8) on reconstruction quality.

In this way, this work brings some original contributions. Starting with the proposal and implementation of a complete encoder/decoder pipeline based on a lightweight CAE custom-made to LoRa constraints; a systematic evaluation of seven latent vector dimensions ($d \in \{8, 16, 32, 64, 128, 256, 512\}$), characterizing the quality–payload trade-off; a quantitative comparison with a PCA baseline; A quantitative comparison with a PCA baseline across all evaluated dimensions, using PSNR and SSIM as reconstruction quality metrics; Identification of the optimal latent dimension under the single-packet LoRa constraint.

The remainder of this paper is organized as follows: Sect. 2 presents the state of the art; Sect. 3 describes the methodology; Sect. 4 discusses the experimental results; and Sect. 5 concludes the paper and present the main future work.

2 State of the Art

The transmission of image data over LoRa networks has been identified as a relevant technical challenge since the emergence of the technology. Chen et al. [16] proposed the Multi-Packet LoRa (MPLR) protocol, designed for reliable transmission of long messages in agricultural monitoring networks. By eliminating per-packet acknowledgement mechanisms typical of conventional Additive Links On-line Hawaii Area (ALOHA)-based approaches and replacing them with selective Automatic Repeat reQuest (ARQ), the protocol reduced image transmission time by 24% to 49% under packet loss rates ranging from 0% to 10%.

Ji et al. [17] introduced a LoRa-based visual monitoring system for agricultural IoT that transmits only modified regions of an image, decomposed into grid patches, making use of the temporal redundancy of static cameras. However, this approach is not generalizable to active scenes or sporadic image acquisition scenarios.

In the paper [18] Staikopoulos et al. presented a comprehensive survey of image transmission approaches over LoRa networks. In this study, they showed that neither the direct fragmentation with ARQ, pre-transmission classical compression, and resolution subsampling satisfies quality and efficiency requirements

under strict bandwidth constraints. Pointing the deep learning-based compression as an encouraging direction. Borsos [19] further confirmed this drawback by comparing JPEG-based approaches under LoRaWAN payload constraints in agricultural scenarios, concluding that the resulting image quality is frequently insufficient for visual diagnostic applications.

Showing that nonlinear learned transformations can outperform JPEG in terms of PSNR at equivalent bitrate, the work of Ballé et al. [14] introduced the first end-to-end optimized variational auto-encoder architecture. Bank et al. [12] present some systematized architectural variants of autoencoders and their application domains, demonstrating that convolutional autoencoders offer advantages over linear methods such as PCA by encoding complex correlation patterns that linear transformations fail to represent efficiently.

In [11], Jamil et al. identified the high computational cost of neural compression, which is normally 100 to 10,000 times greater than JPEG, as a key barrier to industrial adoption, motivating the development of lightweight architectures suitable for embedded deployment. Guerrisi et al. [20] demonstrated the feasibility of deploying neural encoders on severely resource-limited platforms for onboard satellite image compression, achieving PSNR values above 33 dB.

The Tiny Machine Learning (TinyML) paradigm, which refers to the deployment of machine learning models on extremely resource-constrained devices, has emerged as an active research field with exponential growth in publications since 2019 [21]. Capogrosso et al. [22] identified model quantization as one of the most prevalent optimization techniques for reducing memory footprint and inference latency on microcontrollers. Abadade et al. [23] present some use cases in plant disease classification and growth monitoring implemented on microcontrollers. In these experiments, it was shown that lightweight CNN models with INT8 quantization can achieve accuracies above 90% with inference latencies below 100 ms on ESP32 platforms. Also in the context of sensor data compression in embedded systems, Gomes et al. [24] prove that neural autoencoders are a viable solution with inference times compatible with real-time applications on low-end devices.

PCA is widely used as a baseline model for its mathematical foundation and efficient implementation. Based on this, some studies have been carried out, such as Jia et al. [15], who concluded that PCA obtains competitive performance only when the data correlation structure is approximately linear, which is not typically the case for image data. While the leading principal components capture low-frequency structures effectively, they are less efficient in representing high-frequency features and local texture patterns relative to nonlinear representations learned by convolutional autoencoders. This limitation becomes especially noticeable at low latent dimensions ($d \leq 64$), which correspond exactly to the regime of interest for single-packet LoRa transmission.

Based on all of the works, and, despite considerable advances in LoRa-based image transmission, neural image compression, and TinyML, there is, to the best of the authors' knowledge, no work that systematically proposes and evaluates a complete neural encoder/decoder pipeline specifically optimized for LoRa chan-

nel constraints, including encoder deployment on a microcontroller and direct comparison with a PCA baseline across a wide range of latent dimensions. This work intends to address this gap.

3 Methodology

The experimental testing of the proposed pipeline was carried out using the MNIST dataset [25], which consists of 70,000 grayscale images of handwritten digits with an original resolution of 28×28 pixels, 60,000 for training and 10,000 for testing. Although MNIST is traditionally used for classification tasks, its adoption in this work is justified as a controlled validation environment: the dataset exhibits sufficient visual variability to assess the compression and reconstruction capabilities of the models, while its public availability guarantees reproducibility and facilitates direct comparison with future work.

To address the input requirements of the proposed encoder, all images were resized using bilinear interpolation, going from 28×28 to 96×96 pixels, and normalized to the range $[0, 1]$. These dimensions were chosen because of the convolutional architecture, which applies four downsampling operations with a stride of 2, reducing the spatial resolution to 6×6.

As shown in Fig. 1, the architecture is an asymmetric CAE composed of two distinct components: a lightweight encoder designed for execution on limited-resource embedded systems, and a more expressive decoder intended for execution on the receiver server. The encoder consists of four sequential convolutional blocks, each comprising a Conv2D layer with a 3×3 kernel and stride 2 (progressively downsampling the input from 96×96 to 6×6 pixels), followed by Batch Normalization and ReLU activation. The number of filters per block follows the sequence [16, 32, 48, 64], keeping the total number of encoder parameters below 50,000, consistent with ESP32-S3 memory bounds.

To achieve the last vector, a global average pooling layer is used. In this operation, the spatial dimension is reduced from $(6 \times 6 \times 64)$ to a 64-element vector. In the continuity of the process, a fully connected layer without activation allows unconstrained real-valued representations prior to quantization. In the decoding part executed on the server, all of this process is undone. For that, four transposed convolution blocks with stride 2, Batch Normalization, and ReLU activation, followed by a final transposed convolution with Sigmoid activation, are used to reconstruct the image in the range $[0, 1]$.

The asymmetry between encoder and decoder is a purposeful design choice: the encoder omits residual connections to minimize inference memory consumption, while the decoder incorporates them to improve reconstruction quality.

The model was trained using the Adam optimizer with an initial learning rate of 1×10^{-3} and a composite loss function combining Mean Squared Error (MSE) and Structural Similarity (SSIM), defined as:

$$\mathcal{L} = 0.7 \cdot \mathcal{L}_{MSE} + 0.3 \cdot \mathcal{L}_{SSIM}.$$

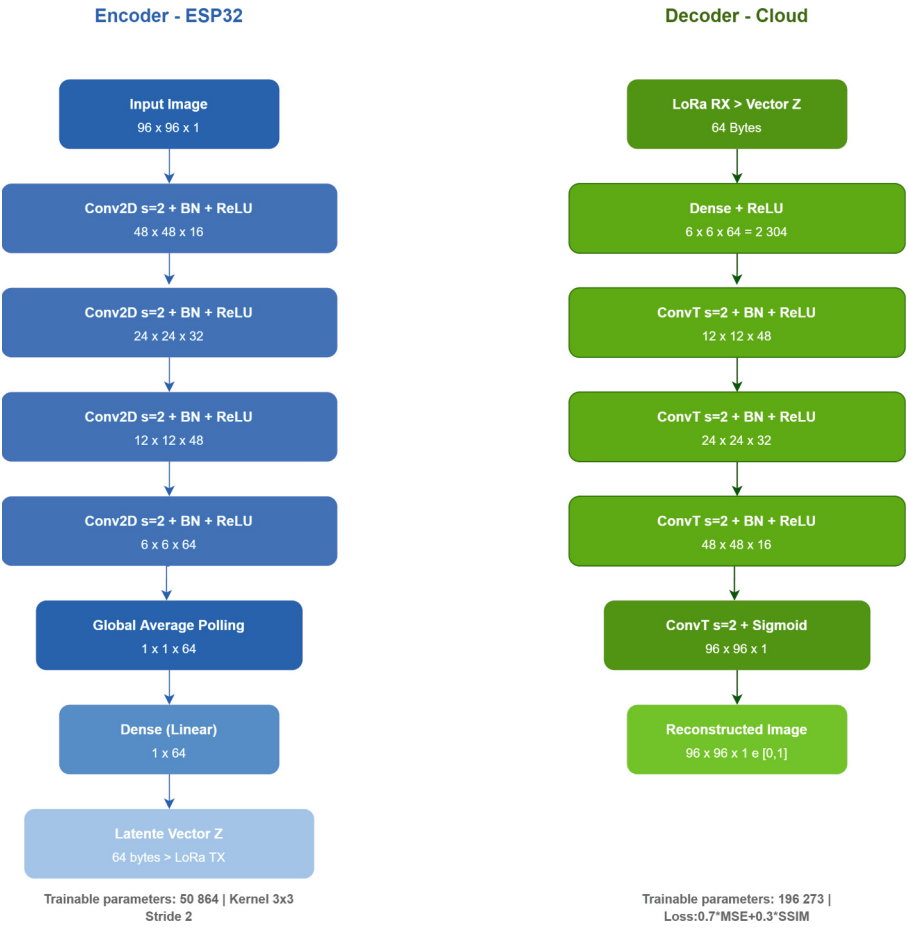

Fig. 1. Asymmetric convolutional autoencoder used in this work.

This formulation balances pixel-wise fidelity with perceptual structure preservation, and was empirically found to produce sharper reconstructions than pure MSE minimization.

All the training hiperparameters are present in the Table 1 namely adaptive early stopping, including a warm-up period of 15 epochs during which stopping criteria were disabled, followed by a patience of 20 epochs and a minimum improvement threshold of 1×10^{-5}. The maximum number of epochs was set to 200. The learning rate was reduced by a factor of 0.5 whenever validation loss did not improve for 8 consecutive epochs (ReduceLROnPlateau). Training was conducted on an NVIDIA RTX 3060 Laptop GPU (6 GB VRAM) with a batch size of 64 and a fixed random seed of 42 to maintain reproducibility.

As mentioned before the baseline adopted for comparison is PCA [15], a widely used linear compression technique. For each evaluated dimension d, PCA

Table 1. Training hyperparameters of the convolutional autoencoder.

Hyperparameter	Value
Optimizer	Adam
Initial learning rate	1×10^{-3}
Loss function	0.7· MSE +0.3· SSIM
Batch size	64
Warm-up (epochs)	15
Early stopping patience	20
Minimum delta	1×10^{-5}
ReduceLROnPlateau factor	0.5
ReduceLROnPlateau patience	8
Maximum epochs	200
Random seed	42

is fitted on the training set after pixel-wise z-score normalization, projecting each image into a vector of d principal components. Reconstruction is obtained by inverse projection followed by denormalization. This comparison allows isolating the effective gain of nonlinear learned representations over linear decompositions at identical payload sizes.

The experimental space includes seven latent vector dimensions $d \in \{8, 16, 32, 64, 128, 256, 512\}$, covering ultra-compact to high-dimensional representations. When the latent vector size exceeds the maximum LoRa payload (approximately 115 bytes under SF9, 125 kHz, 868 MHz), it must be split into multiple fragments, each transmitted in a separate packet. The mapping between latent dimension, number of fragments, and resulting air-time is summarised in Table 2.

Table 2. LoRa transmission configurations for each latent dimension (SF9, 125 kHz, 868 MHz).

d	Payload (bytes)	Fragments	Air-time (s)	Regime
8	8	1	0.52	Single-packet
16	16	1	0.52	Single-packet
32	32	1	0.52	Single-packet
64	64	1	0.52	Single-packet
128	128	2	1.05	Multi-fragment
256	256	3	1.57	Multi-fragment
512	512	5	2.61	Multi-fragment

Prior to transmission, the FP32 latent vector is quantized to UINT8 using min-max normalization based on training statistics, reducing the payload from $4d$ bytes to d bytes, with an additional overhead of 4 bytes for dequantization parameters.

Reconstruction quality is evaluated using PSNR and the Structural Similarity Index Measure (SSIM) [26]. Both metrics are computed between the original input image and the image reconstructed by the decoder, over the full test set (10 000 images), using batches of 256, and reported as mean values. PSNR is computed as:

$$\mathrm{PSNR} = 10 \cdot \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}} \right),$$

where MAX is the maximum possible pixel value (1.0 after normalisation) and MSE is the mean squared error between original and reconstructed images. Values above 25 dB are typically considered acceptable for lossy compression [11]. SSIM evaluates perceptual similarity in terms of luminance, contrast, and structure, ranging from 0 (no similarity) to 1 (identical). For PCA, the cumulative explained variance is additionally reported as an interpretable indicator of retained information.

Both metrics are computed over the full test set (10,000 images), using batches of 256, and reported as mean values. Additionally, for PCA, the cumulative explained variance is reported as an interpretable indicator of retained information.

The LoRa transmission layer was simulated in software, modelling payload constraints, fragmentation protocol (ALFP), and air-time based on physical channel parameters: Spreading Factor (SF), Bandwidth (BW), and Coding Rate (CR). Under the adopted configuration (SF9, BW = 125 kHz, 868 MHz band), the maximum useful payload per packet is approximately 115 bytes. No physical LoRa hardware was used at this stage, as simulation is sufficient to validate the compression pipeline and quantify quality–transmission trade-offs.

Upcoming work will include integration with real hardware (SX1276 and ESP32-S3 with TFLite Micro), verification on agricultural datasets, and measurement of response time and energy consumption in field conditions. All code was implemented in Python 3.10 using TensorFlow 2.12, scikit-learn, and pandas, and will be publicly released upon publication to ensure full reproducibility.

4 Results and Discussion

Table 3 presents the complete experimental results, including PSNR and the SSIM metrics for both the CAE and the PCA baseline, as well as convergence epochs and explained variance for all evaluated latent dimensions. The analysis show three key patterns, the first one is the consistent superiority of the CAE over PCA in the single-packet LoRa regime. The second is the existence of a sweet spot at $d = 64$ that maximizes reconstruction quality per transmitted byte. And the last one is the pronounced diminishing returns beyond $d = 64$,

making multi-fragment configurations of marginal relevance for the proposed system.

Table 3. Complete experimental results. Single-packet LoRa configurations achieving acceptable quality (PSNR $\geq$ 25 dB) are highlighted.

d	CAE				PCA			
	PSNR (dB)	SSIM	Epochs	Frags.	PSNR (dB)	SSIM	Var. Expl. (%)	Frags.
8	20.62	0.849	64	1	15.39	0.505	27.51	1
16	23.28	0.900	70	1	16.71	0.556	38.50	1
32	25.95	0.937	139	1	18.15	0.616	52.13	1
64	**27.74**	**0.955**	137	1	20.12	0.679	68.08	1
128	27.41	0.953	122	2	22.95	0.745	84.21	2
256	27.29	0.950	78	3	27.81	0.832	95.76	3
512	28.35	0.960	137	5	36.16	0.936	99.60	5

Figure 2 provides a visual comparison of the reconstructed images produced by the CAE and PCA at six latent dimensions, from $d = 8$ to $d = 256$, for a representative MNIST sample. The progressive improvement in CAE reconstruction quality is clearly visible, with recognizable digit structure already recovered at $d = 32$ and near-original fidelity achieved at $d = 64$. In contrast, PCA reconstructions remain blurry and structurally degraded across all single-packet configurations, confirming the quantitative results in Table 3.



Fig. 2. Visual comparison of CAE and PCA reconstructions across latent dimensions $d \in \{8, 16, 32, 64, 128, 256\}$ for a single test sample. Green borders indicate single-packet LoRa configurations ($d \leq 64$); red borders indicate multi-fragment configurations ($d > 64$).

The CAE consistently outperforms PCA across all dimensions up to $d = 256$, with the best values observed in the single-packet regime. Looking at the dimension of $d = 32$, its possible to check that the CAE achieves 25.95 dB, compared to 18.15 dB for PCA, representing a 7.8 dB gain and an approximate sixfold reduction in mean squared error. At $d = 64$, the advantage remains significant at 7.6 dB (27.74 dB versus 20.12 dB). This consistent superiority reveals the limitations of linear methods when applied to image data, where pixel correlations are fundamentally nonlinear [12].

Also, the SSIM metric has the same behavior, this is confirmed by the values $d = 64$, where CAE achieves an SSIM of 0.955 compared to 0.679 for PCA, indicating significantly better preservation of perceptual structure at identical compression rates. Notably, PCA at $d = 64$ explains only 68.1% of the total variance, which accounts for its lower reconstruction quality.

With a PSNR of 27.74 dB and an SSIM of 0.955, the dimension of $d = 64$ is the best configuration. This surpasses commonly accepted quality thresholds (PSNR $\geq$ 25 dB, SSIM $\geq$ 0.90) while requiring only 64 bytes (a single LoRa packet). Increasing to $d = 128$ (two fragments) only a substantial improvement of 0.33 dB (+1.2%) while doubling transmission air-time from 0.52 s to 1.05 s and introducing fragmentation overhead. In $d = 256$ the results have a slight degradation of 0.45 dB compared to $d = 64$, suggesting an approach to a suboptimal local minimum, as reflected by the lower number of training epochs (78 vs 137). In $d = 512$, the CAE outperforms $d = 64$ more clearly (+0.61 dB), but the number of LoRa packages increases to five and the transmission time to 2.61 s.

In the Fig. 3, it's possible to see six representative reconstructions at $d = 64$ for both methods. The CAE consistently preserves stroke geometry and contrast in all digit classes, with PSNR values ranging from 25.1 dB to 35.3 dB. PCA reconstructions at the same payload exhibit visible blurring and loss of fine structure, with PSNR values ranging from 17.5 dB to 23.1 dB, underscoring once again the advantage of nonlinear learned representations.

When looking at the last dimension of approximately $d = 512$, there is a reversal of trends; here, the PCA surpasses the CAE, reaching approximately 36.16 dB compared to 28.15 dB. This reversal is essentially expected, as it is a nearly flawless regime for the PCA, explaining about 99.6% of its variance. In the case study, this transmission is practically impossible due to the high fragmentation that would be necessary, as well as the increase in message airtime.

Table 4 summarises the differential performance gain of CAE over PCA across all evaluated dimensions, confirming the consistent advantage of nonlinear representations in the single-packet regime.

Looking at the number of epochs until early stopping its possible to see the complexity of the compression task. Latent dimensions $d = 32$ and $d = 64$ require the largest number of epochs (139 and 137, respectively), indicating the difficulty of learning compact yet informative representations. At Lower dimensions, the convergence tends to be faster due to reduced reconstruction requirements. While at the dimension of $d = 256$ unexpectedly fast convergence, apparently indicating convergence toward a suboptimal local minimum.



Fig. 3. Qualitative reconstruction results at $d = 64$ for six test samples, comparing the proposed CAE with the PCA baseline.

Table 4. Differential performance gain of CAE over PCA.

d	ΔPSNR (dB)	ΔSSIM	Interpretation
8	+5.23	+0.344	CAE superior, low quality
16	+6.55	+0.344	CAE superior, moderate quality
32	+7.81	+0.321	CAE superior, acceptable quality
64	+7.62	+0.276	Optimal trade-off
128	+4.46	+0.207	Reduced CAE advantage
256	−0.52	+0.118	PCA competitive
512	−7.81	+0.024	PCA dominates

This study shows some limitations, starting with the validation. This was carried out exclusively on MNIST, whose grayscale handwritten digit images differ significantly from real agricultural images in terms of texture, illumination variability, and acquisition artifacts. Second, LoRa transmission was simulated rather than experimentally validated with real hardware, and therefore channel effects such as packet loss and noise were not considered. Third, the encoder was not quantized to INT8 nor deployed on an actual ESP32-S3 platform, and thus inference delay and energy consumption were not measured.

5 Conclusion and Future Work

This work proposed and evaluated a CAE-based image compression and reconstruction pipeline specifically designed for transmission over LoRa channels

within resource-constrained embedded systems. The system follows an encode–transmit–decode paradigm: a lightweight encoder compresses the captured image at the sensor node into a low-dimensional latent vector, which is transmitted via LoRa in a single packet or multiple fragments depending on its size, and a decoder at the receiver reconstructs the image from the received representation.

The experimental evaluation, conducted on the MNIST dataset resized to 96×96 pixels, covered seven dimensions ($d \in \{8, 16, 32, 64, 128, 256, 512\}$) and included a comparison with one of the most traditional classical linear compression (PCA), using PSNR along with the SSIM as quality metrics.

The results show that the convolutional autoencoder regularly outperforms PCA across all single-packet transmission configurations ($d \leq 64$), achieving PSNR gains ranging from +5.2 dB ($d = 8$) to +7.8 dB ($d = 32$). In particular, $d = 64$ appears as the optimal operating point of the system. With only 64 bytes of payload transmittable within a single LoRa packet without fragmentation, the neural encoder achieves 27.7 dB PSNR as well as 0.955 SSIM, surpassing commonly accepted quality thresholds (25 dB and 0.90, respectively) [11].

To contextualise these results in terms of practical transmission feasibility, it is instructive to contrast the proposed approach with the uncompressed baseline. A raw 96×96 grayscale image requires 9 216 bytes, which—under the adopted LoRa configuration (SF9, BW = 125 kHz, CR = 4/5, 868 MHz)—necessitates approximately 81 packets and yields an estimated total air-time of approximately 42 s (computed as 81×0.52 s per packet), not accounting for inter-packet gaps or the duty-cycle restrictions imposed by ETSI EN 300 220-2, which would extend the effective wall-clock transmission time considerably. For PCA at $d = 64$, the payload is identical (64 bytes, single packet, 0.52 s air-time), but reconstruction quality is substantially lower (20.12 dB PSNR, SSIM 0.679) than the CAE (27.74 dB PSNR, SSIM 0.955), confirming that the benefit of the CAE lies not in further payload reduction but in superior reconstruction quality at the same transmission cost. By contrast, the proposed CAE at $d = 64$ encodes the image into a single 64-byte packet transmitted in 0.52 s, representing a 144:1 reduction in payload size relative to the raw image. This quantitative comparison demonstrates that, under LoRa constraints, compression is not simply an optimisation strategy but a central enabler of image transmission, and that the proposed pipeline achieves this with reconstruction quality that surpasses accepted perceptual thresholds (27.7 dB PSNR, SSIM 0.955).

Beyond $d = 64$, diminishing returns become evident: increasing to $d = 128$ (two fragments, doubled air-time) yields only a marginal improvement of 0.3 dB, while $d = 256$ results in a slight degradation compared to $d = 64$. These data confirm that non-linear representations learned by convolutional autoencoders are substantially more efficient than linear PCA projections in the low-dimensional regime imposed by LoRa constraints.

The limitations of this study indicate clear directions for future work. First, validation should be extended to real agricultural datasets—such as PlantVillage [27] and in-field acquired images, to evaluate performance under realistic conditions involving color information, texture variability, and illumination changes.

Second, the encoder should be quantized to INT8 using TensorFlow Lite Micro and deployed on actual ESP32-S3 hardware to measure inference latency, energy consumption per frame, and memory footprint in actual scenarios.

Third, integration with physical LoRa hardware (e.g., SX1276 modules) will enable evaluation under real transmission conditions, including packet loss, RSSI variability, and compliance with duty-cycle constraints defined by ETSI EN 300 220-2. Finally, extending the pipeline to RGB images and higher resolutions constitutes a critical step toward real-world deployment in precision agriculture applications, where color information plays a key role in disease diagnosis and crop maturity assessment.

Future work will explore these directions, including a dedicated study on olive maturity classification in real field conditions using the proposed framework.

Acknowledgments. This work was supported by FCT – Fundação para a Ciência e Tecnologia, I.P., through the projects CeDRI (UID/05757/2025, DOI: 10.54499/UID/05757/2025) and UID/PRR/05757/2025 (DOI: 10.54499/UID/PRR/05757/2025), as well as by the Associate Laboratory SusTEC (LA/P/0007/2020, DOI: 10.54499/LA/P/S0007/2020). Additional support was provided by the NORTE2030-FEDER-01182700 project Olive Grove to Fork 4.0.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Use of Artificial Intelligence Tools. Artificial intelligence–based tools were used solely to improve the clarity and grammar of the English writing. Specifically, Grammarly and ChatGPT were employed for language refinement.

References

1. El Khediri, S.: Wireless sensor networks: a survey, categorization, main issues, and future orientations for clustering protocols. Computing **104**(8), 1775–1837 (2022)
2. Fu, Z., Al-Shareeda, M., Ali, M., Manickam, S., Karuppayah, S.: Wireless sensor networks in the internet of things: review, techniques, challenges, and future directions. Indon. J. Electr. Eng. Comput. Sci. **31**, 1190–1200 (2023)
3. Mowla, M.N., Mowla, N., Shahen Shah, A.F.M., Rabie, K.M., Shongwe, T.: Internet of things and wireless sensor networks for smart agriculture applications: a survey. IEEE Access **11**, 145813–145852 (2023)
4. Aldhaheri, L., et al.: Lora communication for agriculture 4.0: opportunities, challenges, and future directions. IEEE Internet Things J. **12**(2), 1380–1407 (2025)
5. De, S., Jalajamony, H.M., Adhinarayanan, S., Joshi, S., Upadhyay, H., Fernandez, R.: Multimedia transmission over lora networks for IoT applications: a survey of strategies, deployments, and open challenges. Sensors **25**(23) (2025)
6. Lavanya, P., Subba Reddy, I.V.: A comprehensive review on lora implementation in IoT application domains. In: AIP Conference Proceedings, vol. 2516, p. 140001 (2022)
7. Chen, C., Luo, J., Xu, Z., Xiong, R., Shen, D., Yin, Z.: Enabling large-scale low-power lora data transmission via multiple mobile lora gateways. Comput. Netw. **237**, 110083 (2023)

8. Semtech Corporation. Sx1276/77/78/79 datasheet: 137 MHz to 1020 MHz low power long range transceiver (2015). Revision 7
9. Yousuf, A., Rochester, E., Ghaderi, M.: A low-cost LoRaWAN testbed for IoT: implementation and measurements, pp. 361–366 (2018)
10. Wallace, G.K.: The jpeg still picture compression standard. Commun. ACM **34**(4), 30–44 (1991)
11. Jamil, S., Piran, M.J., Rahman, M., Kwon, O.-J.: Learning-driven lossy image compression: a comprehensive survey. Eng. Appl. Artif. Intell. **123**, 106361 (2023)
12. Berahmand, K., Daneshfar, F., Salehi, E.S., Li, Y., Xu, Y.: Autoencoders and their applications in machine learning: a survey. Artif. Intell. Rev. **57** (2024)
13. Tschannen, M., Agustsson, E., Lucic, M.: Deep generative models for distribution-preserving lossy compression (2018)
14. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression (2017)
15. Jia, W., Sun, M., Lian, J., Hou, S.: Feature dimensionality reduction: a review. Complex Intell. Syst. **8** (2022)
16. Chen, T., Eager, D., Makaroff, D.: Efficient image transmission using lora technology in agricultural monitoring IoT systems. In: IEEE iThings/GreenCom/CPSCom/SmartData, pp. 937–944 (2019)
17. Ji, M., Yoon, J., Choo, J., Jang, M., Smith, A.H.: Lora-based visual monitoring scheme for agriculture IoT. In: IEEE Sensors Applications Symposium (SAS), pp. 1–6 (2019)
18. Staikopoulos, A., Kanakaris, V., Papakostas, G.A.: Image transmission via lora networks—a survey. In: IEEE International Conference on Image, Vision and Computing (ICIVC) (2020)
19. Borsos, D.: Image transmission with LoRaWAN in agriculture. In: Kovács, T.A., Nyikes, Z., Berek, T., Daruka, N., Tóth, L. (eds.) HCC 2022. Advanced Sciences and Technologies for Security Applications, pp. 235–246. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-47990-8_21
20. Guerrisi, G., Del Frate, F., Schiavon, G.: Convolutional autoencoder algorithm for on-board image compression. In: IEEE IGARSS, pp. 151–154 (2022)
21. Tsoukas, V., Gkogkidis, A., Boumpa, E., Kakarountas, A.: A review on the emerging technology of tinyML. ACM Comput. Surv. **56**(10), 259 (2024)
22. Capogrosso, L., Cunico, F., Cheng, D.S., Fummi, F., Cristani, M.: A machine learning-oriented survey on tiny machine learning. IEEE Access **12**, 23406–23426 (2024)
23. Abadade, Y., Temouden, A., Bamoumen, H., Benamar, N., Chtouki, Y., Hafid, A.S.: A comprehensive survey on tinyML. IEEE Access **11**, 96892–96922 (2023)
24. Gomes, A., et al.: Autoencoders for embedded sensor data compression: a case study on vehicular IoT systems. In: IEEE ETFA (2025)
25. LeCun, Y., Cortes, C., Burges, C.J.: The MNIST database of handwritten digits (1998). http://yann.lecun.com/exdb/mnist/
26. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
27. Hughes, D.P., Salathé, M.: An open access repository of images for computer-based plant disease diagnostics. arXiv preprint arXiv:1511.08060 (2015)