Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 2, 281–296
DOI: 10.61822/amcs-2026-0019

AUTOENCODER–BASED IMAGE REPRESENTATION LEARNING WITH KOLMOGOROV–ARNOLD NETWORKS

ILONA ANNA URBANIAK [a,*], SYLWESTER WIECZOREK [a], JOANNA KOŁODZIEJ [a,b]

[a]Department of Computer Science
Cracow University of Technology
ul. Warszawska 24, 31-155 Kraków, Poland
e-mail: {ilona.urbaniak,joanna.kolodziej}@pk.edu.pl,
sylwek08122000@gmail.com

[b]NASK—National Research Institute
ul. Kolska 12, 01-045 Warsaw, Poland
e-mail: joanna.kolodziej@nask.pl

Autoencoders are widely used for learning compact latent representations of images. While convolutional architectures dominate this area due to their ability to exploit spatial locality, recent developments in neural network design have introduced Kolmogorov–Arnold networks (KANs), which replace fixed activation functions with learnable univariate mappings derived from the Kolmogorov–Arnold superposition theorem. Although KAN-based autoencoders have recently been explored for image representation tasks, existing studies typically focus on limited datasets, fixed KAN formulations, or reconstruction accuracy solely. In this work, we present a KAN-based image autoencoder model that emphasizes representation quality, model complexity, and computational cost. We design and evaluate KAN-based and FastKAN-based autoencoders under strictly matched latent dimensionality constraints and compare them with a convolutional autoencoder baseline across multiple image datasets. Reconstruction quality is assessed using the mean squared error (MSE), which quantifies pixel-wise reconstruction errors, the peak signal-to-noise ratio (PSNR), which measures signal fidelity on a logarithmic scale, and the structural similarity index (SSIM), which reflects perceptual structural similarity. Experimental results demonstrate that KAN-based autoencoders achieve reconstruction quality comparable to convolutional autoencoders of the same representation size, with statistically significant improvements in the PSNR observed in several scenarios, while exhibiting distinct efficiency trade-offs. These findings clarify the practical role of Kolmogorov–Arnold networks in image autoencoder representation learning and highlight them as an alternative to convolutional architectures in low-resolution image settings.

Keywords: autoencoders, representation learning, Kolmogorov–Arnold networks, FastKAN, image reconstruction, SSIM, PSNR.

1. Introduction

Learning compact representations of high-dimensional data is a fundamental problem in applied mathematics and computer science. In image processing, autoencoders are commonly employed to map images into low-dimensional latent spaces from which the original data can be approximately reconstructed.

Convolutional neural networks dominate image autoencoder design due to their ability to exploit spatial locality. However, alternative neural architectures relying on different inductive biases remain relatively unexplored in this context. A notable example of such an architecture is the Kolmogorov–Arnold network (KAN), recently introduced as a function-approximation framework based on the Kolmogorov–Arnold superposition theorem (Liu *et al.*, 2024). Since then, KAN architectures have been adopted across a range of image analysis studies, including convolutional vision architectures for image classification (Bodner *et al.*, 2024; Ouni *et al.*, 2025; Pituková *et al.*, 2025; Yang *et al.*, 2025), remote sensing

*Corresponding author

and hyperspectral image analysis (Cheon, 2024; Firsov *et al*., 2024; Lobanov *et al.*, 2024), as well as semantic segmentation and image enhancement (Ma *et al.*, 2025; Penkin and Krylov, 2025; Yeh and Liou, 2025).

Unlike traditional multilayer perceptrons, the KAN replaces fixed activation functions with learnable univariate functions parameterized by splines or related basis functions, yielding expressive models with compact parameterizations (Liu *et al.*, 2024). Recent work has also explored KAN architectures for representation learning via autoencoders (Moradi *et al*., 2024; Yu *et al*., 2025). Nevertheless, existing studies often report reconstruction accuracy without jointly analyzing model size and inference cost, or without enforcing strict equality of latent dimensionality when comparing against convolutional baselines.

The aim of this work is to verify the usefulness of the KAN architecture for image autoencoder representation learning and to perform a simple comparative analysis with standard convolutional autoencoders under equal latent dimensionality constraints. This study focuses on learning image representations using autoencoders. Dimensionality reduction achieved by the autoencoder bottleneck is interpreted as representation compactness rather than explicit data compression, as no entropy coding or bitstream generation is performed.

Our contributions include the following:

1. definition, implementation, and evaluation of KAN- and FastKAN-based image autoencoder models under a controlled experimental protocol with *strictly matched latent dimensionality* across all compared architectures;
2. a comparative analysis of the proposed KAN–based models against a convolutional autoencoder in terms of computational efficiency, jointly reporting reconstruction fidelity (MSE, PSNR, SSIM), model complexity (parameter count), and computational cost (encoding inference time);
3. a benchmark study across multiple image regimes, including low-resolution natural images (MNIST, grayscale CIFAR-10 at 16×16), a synthetic geometric Shapes dataset, and higher-resolution natural images (STL-10 resized to 64×64), highlighting statistically validated scenarios in which architectures based on KANs are most effective;
4. an analysis of the relationship between reconstruction quality and computational efficiency in spline-based KANs and the accelerated FastKAN variant, clarifying practical implications for low-latency representation learning;
5. a statistically rigorous evaluation of reconstruction performance using paired nonparametric tests with multiple-comparison correction, enabling reliable identification of scenarios in which KAN-based autoencoders significantly outperform convolutional baselines.

The rest of the paper is organized as follows. Section 2 introduces the theoretical foundations of image autoencoders and Kolmogorov–Arnold networks. Section 3 describes the developed KAN-based autoencoder architectures and training procedure. Section 4 details the experimental setup, datasets, and evaluation metrics. Section 5 presents and discusses the experimental results, with a simple comparative analysis of the KAN-based model and convolutional autoencoders, supported by statistical significance testing. Section 6 concludes the paper and outlines directions for future research.

2. Theoretical background

2.1. Image representation and autoencoders. In autoencoder-based representation learning, a digital image is modeled as a finite-dimensional vector in high-dimensional Euclidean space (Goodfellow *et al*., 2016).

Definition 1. (*Image of spatial resolution*) Let

$$\mathcal{X} \subset \mathbb{R}^{H \cdot W \cdot C} \tag{1}$$

be a subset of high-dimensional Euclidean space. An element of this space $X \in \mathcal{X}$ is an image of spatial resolution $H \times W$ with C channels.

Definition 2. (*Autoencoder learning*) The autoencoder learning procedure is defined by the following mapping:

$$\mathcal{E} : \mathbb{R}^{H \cdot W \cdot C} \to \mathbb{R}^{K}, \tag{2}$$

where $K \ll H \cdot W \cdot C$. The decoder defined by

$$\mathcal{D} : \mathbb{R}^{K} \to \mathbb{R}^{H \cdot W \cdot C} \tag{3}$$

approximately reconstructs the original input, i.e.,

$$\hat{X} = \mathcal{D}(\mathcal{E}(X)) \approx X. \tag{4}$$

Details of formal modelling of autoencoders are presented by Hinton and Salakhutdinov (2006).

From a functional perspective, the encoder–decoder pair defines a composition of mappings that approximates the identity on $\mathcal{X}$ while passing through a lower-dimensional latent space (Bengio *et al*., 2013). Thus, representation learning can be interpreted as the approximation of a high-dimensional function by a composition of functions acting on lower-dimensional domains, as illustrated in Fig. 1.

The above representation refers to the Kolmogorov–Arnold (K–A) superposition theorem (Kolmogorov, 1957; Arnold, 1957).

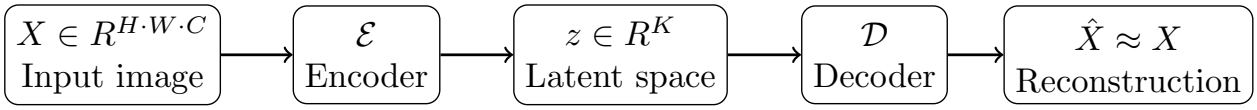

Fig. 1. Autoencoder-based image representation learning as a composition of mappings $\mathcal{D} \circ \mathcal{E}$ approximating the identity on image space.

Theorem 1. (Kolmogorov–Arnold superposition theorem) *Let*

$$f : [0,1]^n \to \mathbb{R} \tag{5}$$

be a continuous multivariate real function. Then f can be represented as the following finite composition of functions of a single variable:

$$f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \varphi_{q,p}(x_p) \right), \tag{6}$$

where $\varphi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$.

The proof of (K–A) can be found in the original works of Kolmogorov and Arnold (Kolmogorov, 1957; Arnold, 1957), with a constructive proof given by Braun and Griebel (2009).

In the context of representation learning, the complex transformations from image space to latent space are realized through structured compositions of adaptive univariate mappings. Kolmogorov–Arnold networks build upon this principle by replacing fixed activation functions with learnable univariate functions, enabling flexible approximation of high-dimensional relationships within compact network architectures (Liu *et al.*, 2024). Later refinements demonstrated that such representations can be further simplified under mild conditions (Lorentz, 1966).

2.2. KAN architectural model. In conventional multilayer perceptrons (MLPs), each neuron computes an affine transformation followed by a fixed nonlinear activation function, such as ReLU or sigmoid (Goodfellow *et al.*, 2016). This design has proven effective in a wide range of representation learning tasks (Bengio *et al.*, 2013; Hinton and Salakhutdinov, 2006). The network's expressive power is largely determined by the number of neurons and layers, rather than by the adaptability of the activation functions themselves.

Figure 2 illustrates the conceptual structure of a Kolmogorov–Arnold network. In contrast to conventional neural architectures, where nonlinearity is applied at the level of neurons via fixed activation functions, the KAN associates each edge with a learnable univariate function. The nodes perform summation operations, yielding compositions of the form prescribed by the Kolmogorov–Arnold superposition theorem.

Lower layers apply collections of univariate transformations to individual scalar inputs, while higher ones aggregate these responses through summation and further univariate mappings. As a result, the overall network realizes a high-dimensional mapping through a hierarchical composition of adaptive one-dimensional functions. This explicit separation between summation and nonlinear transformation distinguishes KANs from standard multilayer perceptrons and provides a direct architectural interpretation of the underlying approximation-theoretic principles.

Kolmogorov–Arnold networks, introduced by Liu *et al.* (2024), depart from this paradigm by replacing fixed activation functions with learnable univariate functions. In a KAN layer, each edge is associated with a univariate function that acts on a scalar input, and these functions are optimized during training. The resulting architecture directly reflects the structure suggested by the Kolmogorov–Arnold superposition theorem (Kolmogorov, 1957; Arnold, 1957), which states that multivariate continuous functions can be represented as finite compositions and summations of univariate functions.

In practice, the learnable univariate functions in a KAN are parameterized using spline bases or related function expansions (Liu *et al.*, 2024). This parameterization allows the network to adapt the shape of the nonlinearity to the data, rather than relying on a predefined functional form. Classical results in approximation theory indicate that such adaptive basis representations can achieve high expressive power with relatively compact models (Lorentz, 1966).

KAN architectures have been explored in a growing number of image analysis tasks. Extensions incorporating a convolutional structure have been proposed to better exploit spatial locality in images (Bodner *et al.*, 2024; Ouni *et al.*, 2025), and hybrid architectures combining KANs with capsule networks or other modules have been studied in medical image classification (Pituková *et al.*, 2025; Yang *et al.*, 2025). KAN-based models have also been applied to remote sensing and hyperspectral images analysis (Cheon, 2024; Firsov *et al.*, 2024; Lobanov *et al.*, 2024; Ma *et al.*, 2025), as well as image enhancement and segmentation tasks (Penkin and Krylov, 2025; Yeh and Liou, 2025).

KAN-based autoencoder architectures have been

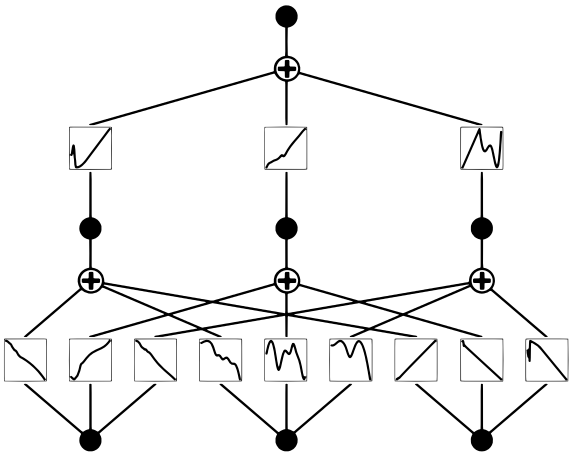

Fig. 2. Illustrative structure of a Kolmogorov–Arnold network. Each edge is associated with a learnable univariate function, while nodes sum incoming signals. The overall mapping is realized as a structured composition and summation of adaptive one-dimensional functions, reflecting the Kolmogorov–Arnold superposition principle.

recently investigated for representation learning (Moradi *et al.*, 2024). These studies demonstrate the feasibility of using KAN layers within encoder–decoder structures, but differ in architectural choices, evaluation protocols, and the scope of comparative analysis. In contrast, the focus of this work is on systematically evaluating the KAN and FastKAN architectures as components of image autoencoders under controlled latent dimensionality constraints, with a direct comparison to a convolutional autoencoder baseline.

2.3. KAN layer formulation and properties.

Definition 3. (*KAN layer*) A single KAN layer can be defined as the following mapping of an input vector $x = [x_1, \ldots, x_d] \in \mathbb{R}^d$ to an output vector $y = [y_1, \ldots, y_{d'}] \in \mathbb{R}^{d'}$:

$$y_i = \sum_{j=1}^{d} \varphi_{i,j}(x_j), \qquad i = 1, \ldots, d', \tag{7}$$

where each $\varphi_{i,j} : \mathbb{R} \to \mathbb{R}$ is a learnable univariate function associated with the edge connecting input x_j to the output neuron y_i.

Although the functions $\varphi_{i,j}$ can be arranged in a matrix form, this structure does not define a linear operator, since the resulting mapping is not linear in its input.

In the above definition, the role of scalar weights in a standard fully connected layer is replaced by adaptive function mappings.

Definition 4. (*Multilayer KAN*) An L-layer KAN network is defined as a recursive composition of L layers,

$$h^{(0)} = x, \quad h^{(l+1)} = \Phi_l\Big(h^{(l)}\Big), \quad l = 0, 1, \ldots, L-1, \tag{8}$$

where Φ_l denotes the transformation implemented by the l-th KAN layer.

The conventional multilayer perceptron (MLP) layer is usually defined by the following matrix model:

$$h^{(l+1)} = \sigma\Big(W_l h^{(l)} + b_l\Big), \tag{9}$$

with a fixed activation function σ and a matrix of scalar weights W_l.

In KANs, the matrix W_l is replaced by a matrix of univariate functions $\{\varphi_{i,j}\}$ (see Eqn. (7)). In practical implementations, $\varphi_{i,j}$ are parameterized using flexible basis expansions, most commonly cubic B-splines (Liu *et al.*, 2024). Such representations provide strong approximation capabilities while preserving locality, as changes in spline parameters affect only a limited region of the input domain (Lorentz, 1966). As a result, each connection in a KAN layer typically carries multiple parameters, leading to a higher parameter count per connection than in standard MLPs.

An important property of KANs is interpretability. Since each $\varphi_{i,j}$ is a univariate mapping, they can be directly visualized and analyzed after training. This contrasts with conventional neural networks, where nonlinear transformations are distributed across layers and are more difficult to interpret (Liu *et al.*, 2024).

In a KAN, evaluating a nonlinear function on each connection is more expensive than a scalar multiplication. To mitigate this, accelerated variants such as FastKAN replace spline-based functions with radial basis function approximations, achieving faster inference while preserving the core architectural principles of KAN (Liu *et al.*, 2024).

3. KAN-based autoencoders

3.1. Architecture overview. The proposed KAN-based autoencoder follows the standard encoder–decoder paradigm, with KAN layers replacing classical fully connected or convolutional layers in the latent mapping. The overall architecture consists of three main components: an encoder $\mathcal{E}$, a latent bottleneck representation, and a decoder $\mathcal{D}$.

Definition 5. (*KAN-based autoencoder*) Given an input image $X \in \mathbb{R}^{H \cdot W \cdot C}$, the encoder maps the flattened image vector to a latent representation,

$$z = \mathcal{E}(X) \in \mathbb{R}^K, \tag{10}$$

where $K \ll H \cdot W \cdot C$. The decoder then reconstructs the image via

$$\hat{X} = \mathcal{D}(z). \tag{11}$$

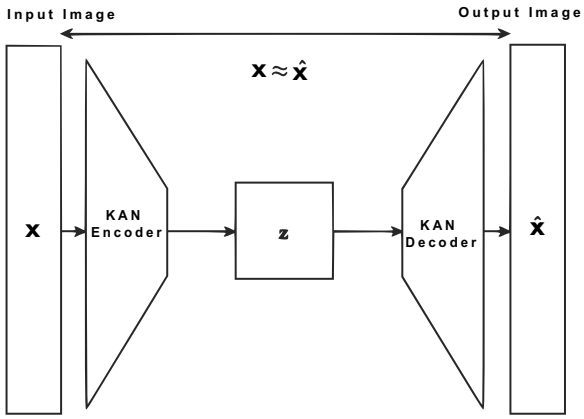

Fig. 3. Architecture of the proposed KAN-based autoencoder. The encoder and decoder are implemented as KAN layers, mapping the flattened input image x to a latent representation z and reconstructing the output $\hat{x}$.

The conceptual KAN-based autoencoder architectural model is presented in Fig. 3.

In contrast to convolutional autoencoders, which rely on spatial locality and shared weights, the encoder and decoder in the proposed model are constructed using stacked KAN layers. Each performs a summation of learnable univariate transformations, as described in Section 2.3, enabling flexible nonlinear mappings between successive feature representations.

The encoder progressively reduces dimensionality by applying a sequence of KAN layers of decreasing width, culminating in the latent bottleneck. The decoder mirrors this structure symmetrically, expanding the latent representation back to the original image dimensionality. The latent dimensionality is selected such that representation size is identical across all compared models, ensuring a fair evaluation against the convolutional autoencoder baseline.

To investigate computational trade-offs, two variants of the architecture are considered. For clarity, throughout the results section, we refer to the standard spline-based KAN variant as KAN-linear to distinguish it from FastKAN. This label is used only as a naming convention for the baseline KAN implementation adopted in our experiments and does not imply that the resulting mapping is linear in the strict algebraic sense. The KAN-linear variant employs spline-based KAN layers, closely following the original formulation. FastKAN replaces spline parameterizations with Gaussian radial basis functions, yielding a FastKAN-based autoencoder that reduces computational cost while preserving the core architectural principles.

3.2. Training procedure. All autoencoder models are trained in an unsupervised manner by minimizing the reconstruction error between the input image X and its reconstruction $\hat{X}$. The primary loss function is the mean squared error (MSE),

$$\mathcal{L}_{\mathrm{MSE}} = \|X - \hat{X}\|_2^2, \tag{12}$$

which encourages accurate pixel-wise reconstruction.

Input images are normalized to the interval $[0, 1]$ prior to training. Model parameters, including spline coefficients or radial basis function parameters in KAN layers, are optimized using gradient-based optimization. Training is performed using a mini-batch stochastic gradient descent with an adaptive optimizer, and learning rates are selected empirically to ensure stable convergence.

To prevent overfitting and improve generalization, early stopping based on validation loss is employed. All models are trained under identical conditions, including batch size, the number of training epochs, and latent dimensionality. This controlled setup ensures that the observed performance differences can be attributed to architectural choices rather than to variations in training protocols.

In addition to the MSE, reconstruction quality is evaluated using the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM), providing complementary objective and perceptual measures of reconstruction fidelity. Inference time and model size are also recorded to assess the practical efficiency of KAN-based autoencoders relative to the convolutional baseline.

Due to the functional nature of KAN layers, several KAN-specific considerations were incorporated into the training procedure. In each KAN layer, the only trainable parameters are the coefficients of the univariate basis functions associated with network edges; no scalar weights are used.

The univariate functions $\varphi_{i,j}$ were parameterized using cubic B-splines defined on a fixed, deterministic grid spanning the interval $[-5, 5]$. Spline coefficients were initialized randomly according to a Gaussian distribution,

$$\text{coeffs} \sim \mathcal{N}(0, 0.05^2), \tag{13}$$

which ensures that the initial magnitude of the learned functions is small. As a result, both encoder outputs and decoder inputs are initially close to zero, providing a stable starting point for training. During optimization, the coefficients are progressively updated, enabling the learned functions to deviate from linearity and increase the model's expressive power.

Training was performed using mini-batch stochastic gradient descent with a batch size of 64. The Adam optimizer was employed with an initial learning rate of

$\alpha = 10^{-3}$. All models were trained for 10 epochs; empirical observations showed that the reconstruction error stabilized after several epochs. After each epoch, the reconstruction loss on the validation set was evaluated, and the model parameters yielding the lowest validation error were retained for final evaluation.

Finally, although all models are trained purely as autoencoders using a reconstruction objective, the presence of an explicit low-dimensional bottleneck enables the learned latent representation to be interpreted as a compact encoding of the input image. By varying the latent dimensionality, the reconstruction task becomes progressively more constrained, enabling a systematic evaluation of how different architectural choices affect reconstruction fidelity as representational compression increases.

4. Experimental setup

4.1. Datasets. To evaluate the behavior of KAN-based autoencoders across different data characteristics, experiments were conducted on several image datasets, including both natural and synthetic images, as listed below.

CIFAR-10. The CIFAR-10 dataset (Krizhevsky, 2009) consists of low-resolution natural images originally sized 32×32 pixels. In our experiments, images were converted to grayscale using a luminance transformation and downsampled to 16×16 pixels. For visualization purposes, images are displayed at a higher resolution than the original, leading to visible pixelation.

MNIST. The MNIST dataset (LeCun *et al.*, 1998) contains 70,000 grayscale images of handwritten digits with a resolution of 28×28. Due to its low structural complexity (a uniform background and simple foreground shapes), MNIST serves as a baseline test case, allowing verification that the autoencoder achieves near-perfect reconstruction for a given bottleneck width K.

Shapes. A synthetic dataset called Shapes was generated to assess the ability of KAN models to encode images with sharp, well-defined structures. The dataset consists of simple geometric shapes (circles, squares, triangles, and lines) drawn randomly on a black background. A total of 1,000 32×32 images were generated with random parameters such as position, size, and rotation. These data allow us to investigate whether KAN-based autoencoders can efficiently encode images with an explicit geometric structure. Because this synthetic dataset contains 1,000 procedurally generated images, it is used here as a controlled structural testbed rather than as a standalone benchmark for broad generalization.

Large (STL-10). To evaluate performance on higher-resolution and more diverse imagery, experiments were conducted on grayscale images resized to 64×64 pixels from the STL-10 dataset (Coates *et al.*, 2011). Throughout the paper, this setting is referred to as Large (STL-10), and the terms Large and STL-10 are used interchangeably. The original STL-10 images are 96×96 color photographs; for our experiments, they were converted to grayscale and rescaled. This dataset was selected to test how autoencoders handle high visual variability across natural scenes, textures, and object details.

4.2. Experimental environment and implementation. All experiments were implemented in Python using the PyTorch framework (version 1.13) (Paszke *et al.*, 2019). The code was executed on a MacBook Pro equipped with an Apple M2 Pro processor.

KAN layers were implemented based on the conceptual design of `KolmogorovArnoldLayer` from the `pyKAN` library, but a custom implementation was developed to better suit the requirements of the experiments. The implemented `KANLayer` module inherits from `nn.Module` and initializes all parameters defining the univariate functions $\varphi_{i,j}$ within a given layer.

Spline coefficients are stored in a tensor `coeffs`, representing function values at predefined grid nodes. A deterministic, uniform grid called `self.grid` spanning the interval $[-5, 5]$ is used, and spline interpolation is applied to evaluate the functions. This construction ensures that each input–output connection learns its own nonlinear univariate function, enabling flexible modelling of nonlinear relationships.

From a computational perspective, training time varied significantly across models. Convolutional autoencoders and FastKAN-based autoencoders trained substantially faster than spline-based KAN ones. For example, training for 10 epochs on CIFAR-10 required approximately 45 minutes for the CNN and FastKAN models, whereas the spline-based KAN autoencoder required approximately 3–4 hours, primarily due to the cost of evaluating spline functions during each training iteration.

4.3. Model parameterization and training configuration. To study the effect of compression strength, experiments were conducted for multiple bottleneck widths,

$$K \in \{256, 128, 64, 32, 16\}. \tag{14}$$

For images of size 32×32 ($N = 1024$), the compression ratio $\frac{N}{K}$ ranges from 4 ($K = 256$) to 64 ($K = 16$). For images of size 64×64 ($N = 4096$), the compression ratio ranges from 16 to 256.

Each K configuration was trained and evaluated independently. All models were trained under identical conditions: 10 training epochs per dataset, the Adam

optimizer with a learning rate of 10^{-3}, mean squared error loss, and a batch size of 64. After training, the model parameters corresponding to the lowest validation loss were selected for final evaluation.

4.4. Evaluation metrics. Reconstruction quality between the original input image X and its reconstruction $\hat{X}$ was quantitatively assessed using the mean squared error, peak signal-to-noise ratio, and structural similarity index.

The mean squared error is defined as

$$\mathrm{MSE}(X, \hat{X}) = \frac{1}{H \cdot W \cdot C} \sum_{i=1}^{H \cdot W \cdot C} \left(X_i - \hat{X}_i \right)^2. \quad (15)$$

The peak signal-to-noise ratio is expressed in decibels as

$$\mathrm{PSNR}(X, \hat{X}) = 10 \log_{10} \left(\frac{L^2}{\mathrm{MSE}(X, \hat{X})} \right), \quad (16)$$

where L denotes the maximum possible pixel intensity value.

The structural similarity index measures perceptual similarity between X and $\hat{X}$ by jointly comparing luminance, contrast, and structural components. The structural similarity index is defined as in the work of Wang *et al*. (2004):

$$\mathrm{SSIM}(X, \hat{X}) = \frac{(2\mu_X \mu_{\hat{X}} + c_1)(2\sigma_{X\hat{X}} + c_2)}{(\mu_X^2 + \mu_{\hat{X}}^2 + c_1)(\sigma_X^2 + \sigma_{\hat{X}}^2 + c_2)}, \quad (17)$$

where μ_X, $\mu_{\hat{X}}$ denote mean intensities, σ_X^2, $\sigma_{\hat{X}}^2$ are the corresponding variances, $\sigma_{X\hat{X}}$ stands for the covariance between X and $\hat{X}$, and c_1, c_2 are small constants introduced for numerical stability.

Although the PSNR is a logarithmic transformation of the MSE, we report both metrics to facilitate comparison with prior image reconstruction studies. These metrics are widely used in practical image quality evaluation, particularly in the contexts of compression and medical imaging (Urbaniak and Pinto, 2023). Furthermore, their relationship to diagnostic quality has been analyzed in clinical imaging studies (Kowalik-Urbaniak *et al.*, 2015).

In addition to reconstruction quality, computational aspects of autoencoder models were also evaluated. Computational efficiency was assessed using the encoding inference time, defined as the elapsed time required to compute the latent representation

$$z = \mathcal{E}(X), \quad (18)$$

measured in milliseconds per image under identical hardware and software conditions.

Model complexity was quantified by the total number of trainable parameters in the encoder–decoder architecture.

This evaluation protocol enables a structured comparison of KAN-based autoencoders and convolutional baselines across different data regimes, image resolutions, and compression levels. The reported inference-time results complement this comparison by providing a practical indication of runtime behavior, although their interpretation should account for differences in architectural design and parameterization between models.

4.5. Statistical analysis. To assess whether the observed performance differences between model families are statistically significant, we applied the Wilcoxon signed-rank test to paired validation results. Pairing was performed across datasets and bottleneck widths, i.e., for each dataset and each latent dimensionality K, the metric value obtained by a KAN-based model was paired with the corresponding value from the CNN baseline under the same experimental configuration. This nonparametric test was selected as an alternative to the paired t-test, as it does not assume normality of the paired differences.

Holm correction was subsequently applied to the resulting p-values in order to account for multiple pairwise comparisons within each evaluation metric and to control the family-wise error rate at $\alpha = 0.05$. Statistical analyses were conducted separately for each metric (MSE, PSNR, SSIM) using final validation values.

5. Experimental results

This section presents a comprehensive evaluation of the proposed KAN-based autoencoders, compared with convolutional autoencoders (CNNs) and the FastKAN variant. Results are reported for four datasets of increasing structural complexity: MNIST (28×28), CIFAR-10 (16×16), Shapes (32×32), and Large (64×64). Performance is assessed across five bottleneck widths ($K \in \{16, 32, 64, 128, 256\}$) using quantitative reconstruction metrics (MSE, PSNR, SSIM), inference time, and qualitative visual inspection.

5.1. Quantitative analysis. Table 1 summarizes the best-performing models for each dataset and bottleneck size, selected according to a hierarchical criterion favoring SSIM, followed by the PSNR and MSE. The lower MSE and higher PSNR/SSIM indicate better reconstruction fidelity.

It is important to note that the inference times reported in Table 1 correspond to the selected best-performing model within each configuration. As a result, these values provide a practical indication of

Table 1. Best-performing autoencoder configuration for each dataset and bottleneck width K. Models are selected using a hierarchical criterion prioritizing the highest structural similarity (SSIM), followed by the highest peak signal-to-noise ratio (PSNR), and the lowest mean squared error (MSE). Reported values correspond to the selected best model in each setting. Parameters denote the total number of trainable parameters of the model. Inference time is measured as the average encoding time per image, reported in milliseconds. The reported inference times, therefore, correspond to the best-performing model in each configuration and are not intended as a fully controlled comparison across architectures. Lower values of the MSE and inference time indicate better performance, while higher values of the PSNR and SSIM indicate higher reconstruction quality.

Dataset	K	Best model	MSE ↓	PSNR ↑	SSIM ↑	Parameters	Inference (ms) ↓
MNIST	16	CNN	8.5390	20.5086	0.8852	165,841	0.4304
	32	CNN	5.8139	22.1580	0.9208	217,057	0.4299
	64	FastKAN	3.2175	24.6805	0.9293	905,728	0.0168
	128	FastKAN	1.7049	27.3376	0.9441	1,809,088	0.0158
	256	FastKAN	1.6060	28.0510	0.9430	3,615,808	0.0308
CIFAR-10 (16×16)	16	KAN (linear)	2.6822	20.5737	0.6887	49,152	0.0283
	32	KAN (linear)	1.6254	22.7102	0.8227	98,304	0.1147
	64	KAN (linear)	0.9101	25.2013	0.9049	196,608	0.1072
	128	KAN (linear)	0.5314	27.4981	0.9474	393,216	0.2017
	256	KAN (linear)	0.3496	29.2584	0.9653	786,432	0.3346
Shapes	16	FastKAN	74.9681	11.8792	0.0882	298,048	0.0141
	32	FastKAN	45.4278	14.0272	0.1654	593,008	0.0184
	64	FastKAN	28.6465	15.8047	0.4814	1,182,928	0.0211
	128	FastKAN	23.3089	16.6445	0.5810	2,362,768	0.0382
	256	FastKAN	21.1224	17.2391	0.6490	4,722,448	0.1090
STL-10 (gray, 64×64)	16	CNN	112.5016	16.2021	0.2476	584,977	2.6375
	32	CNN	100.5952	16.6742	0.2709	718,113	2.6351
	64	CNN	107.7177	16.3439	0.2568	984,385	2.6509
	128	CNN	99.2994	16.7280	0.2742	1,516,929	2.6126
	256	FastKAN	80.1732	17.4895	0.2789	18,887,440	0.1374

runtime under the preferred setting for each case, but they do not constitute a fully controlled benchmark for a direct comparison across architectures.

MNIST. For small bottlenecks ($K = 16$ and $K = 32$), the CNN achieves slightly higher SSIM values, reflecting its ability to preserve the global structure of simple digit shapes under strong compression. However, as the latent dimensionality increases ($K \geq 64$), FastKAN consistently outperforms the CNN, reaching SSIM values up to 0.944 and a PSNR above 28 dB while simultaneously reducing the MSE. This behavior indicates that FastKAN more effectively exploits additional latent capacity, enabling sharper reconstructions with a lower reconstruction error.

CIFAR-10. Across the full range of bottleneck widths, the linear KAN autoencoder achieves the best quantitative performance on CIFAR-10. At $K = 256$, it reaches an SSIM of 0.965 and a PSNR of 29.26 dB, outperforming both the CNN and FastKAN. These results suggest that the flexible univariate functions employed in KAN layers are particularly well suited to modelling the local statistics of small, diverse natural images, leading to highly accurate reconstructions even at moderate compression ratios.

Shapes. For the synthetic Shapes dataset, FastKAN dominates in all configurations. SSIM improves monotonically with an increasing bottleneck size, rising from 0.088 at $K = 16$ to 0.649 at $K = 256$. The CNN performs poorly on this dataset, exhibiting low SSIM values even at higher bottlenecks. The superior performance of FastKAN reflects its ability to preserve sharp edges and simple geometric structures, which are poorly captured by convolutional smoothing operations.

Large (STL-10). On the Large dataset, the CNN achieves a marginally better SSIM for smaller bottlenecks ($K = 16$–128), with values around 0.25–0.27. However, at the largest bottleneck ($K = 256$), FastKAN overtakes the CNN, achieving a higher SSIM and PSNR while reducing

the MSE by a substantial amount. This result indicates that, for high-resolution, structurally complex images, KAN-based models require sufficient latent capacity to fully leverage their representational advantages.

5.2. Training dynamics. Figure 4 illustrates the evolution of the MSE, PSNR, and SSIM during training on the MNIST dataset for selected bottleneck sizes. All models exhibit rapid convergence within the first few epochs, followed by slower, more gradual improvements as training progresses. This behavior indicates stable optimization across architectures and evaluation metrics.

Across all datasets (Figs. 4–7), similar training dynamics can be observed. Regardless of dataset complexity, all models converge quickly during the initial training phase, after which performance improvements become more incremental. This pattern is consistent across the MSE, PSNR, and SSIM, as well as different bottleneck dimensionalities, suggesting robust and well-behaved optimization for all autoencoder architectures considered.

CNN-based autoencoders typically exhibit faster initial convergence, particularly for larger bottleneck sizes. However, their performance tends to saturate early, especially under stronger compression (small K), where further reductions in the MSE and improvements in the PSNR and SSIM become limited. This behavior suggests that conventional convolutional encoders efficiently capture a coarse image structure but struggle to refine reconstructions when latent capacity is constrained.

In contrast, KAN-based models demonstrate a more gradual yet sustained improvement throughout training. Both KAN-linear and FastKAN continue to reduce the reconstruction error and improve perceptual quality beyond the early epochs, with this effect being most pronounced for smaller bottleneck sizes. This indicates that, while CNNs quickly capture a global structure, KAN-based architectures are more effective at progressively refining fine-scale details as training proceeds.

Among the evaluated models, FastKAN consistently achieves the lowest final MSE and the highest PSNR across datasets and bottleneck configurations, indicating superior reconstruction fidelity under compression. Although increasing the bottleneck dimensionality improves performance for all models, KAN-based autoencoders exhibit a stronger ability to preserve reconstruction quality at lower latent dimensions. This suggests that the functional representations learned by KANs enable more efficient use of latent capacity compared to standard convolutional encoders.

Although these trends are visually consistent across datasets, training curves alone do not establish whether the observed performance differences are statistically significant. To quantitatively assess these differences and account for variability across datasets and bottleneck configurations, a formal statistical analysis is presented in the following subsection.

5.3. Quantitative statistical analysis. To quantify the statistical significance of differences between the CNN baseline and KAN-based models, we report the results of the Wilcoxon signed-rank tests with Holm correction in Table 2. The statistical protocol is described in Section 4.5. Table the summarizes the Wilcoxon signed-rank test results for MSE, PSNR, and SSIM across datasets and bottleneck sizes.

The results indicate that FastKAN achieves statistically significant improvements over the CNN baseline in terms of both the MSE and PSNR. In particular, the improvement of the latter is highly significant, reflecting consistently higher reconstruction fidelity across datasets and latent dimensions. KAN-linear also demonstrates a statistically significant improvement in the PSNR compared to the CNN baseline, although the corresponding improvement in the MSE does not reach statistical significance after Holm correction.

For SSIM, KAN-based models exhibit higher median values compared to the CNN baseline across most datasets and bottleneck sizes; however, these differences do not remain statistically significant after Holm correction. This suggests that improvements in structural similarity, while present, are more subtle and exhibit greater variability than error-based and signal-to-noise measures.

Although the PSNR is a monotonic logarithmic transformation of the MSE, the Wilcoxon signed-rank test is applied to paired differences after transformation. Consequently, the ranking of absolute paired differences may differ between the MSE and PSNR representations. The logarithmic scaling of the PSNR compresses large error ranges and emphasizes relative differences at lower error levels, which can lead to a different distribution of paired differences and, therefore, different Wilcoxon statistics and corrected significance levels.

Overall, the statistical analysis confirms that KAN-based architectures, particularly FastKAN, provide robust, statistically significant gains in reconstruction accuracy across error magnitude and signal fidelity, while maintaining competitive structural similarity.

5.4. Qualitative analysis. Representative reconstructions for CIFAR-10 are shown in Fig. 8 for a representative bottleneck size ($K = 32$). Under strong compression, CNN reconstructions appear smoother and blur fine details, preserving a coarse structure but degrading texture and edge sharpness. The KAN produces sharper outputs with richer local detail, though slight reconstruction artifacts may appear in some cases.

FastKAN achieves a balance between the two

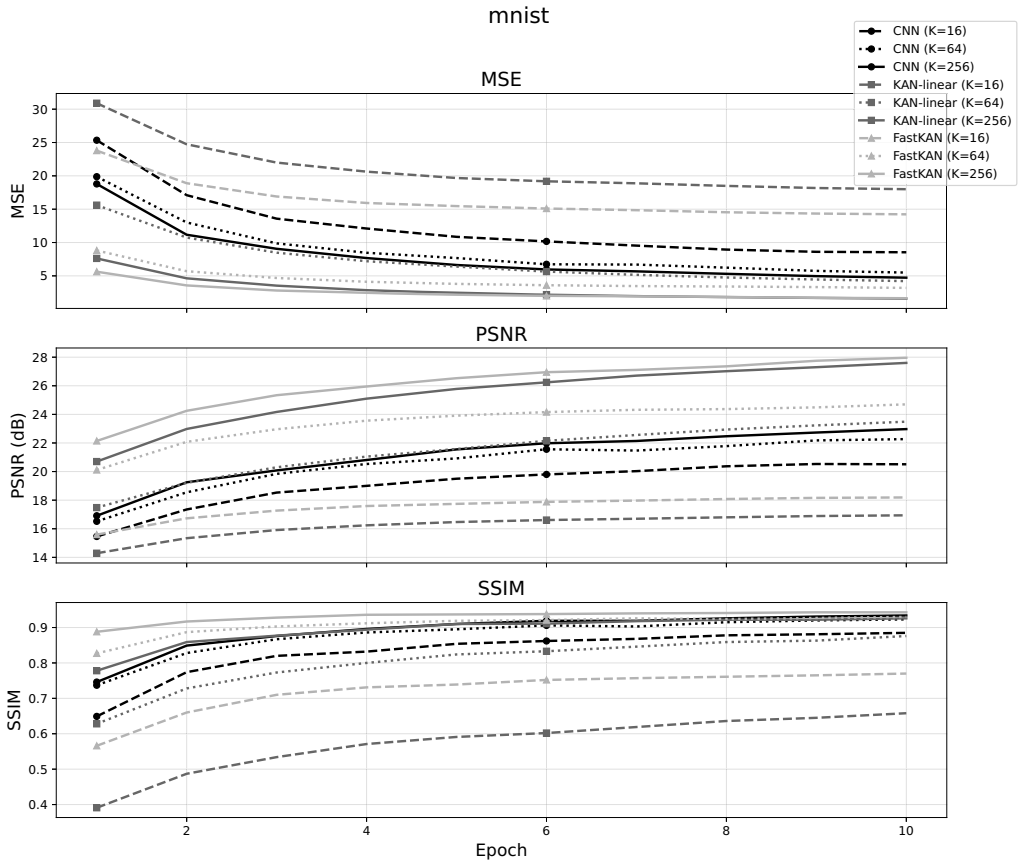

Fig. 4. Training dynamics on MNIST for selected bottleneck widths K. Evolution of the MSE, PSNR, and SSIM across epochs for the CNN, KAN-linear, and FastKAN. For clarity, three bottleneck sizes are shown: $K = 256$ (solid), $K = 64$ (dotted), and $K = 16$ (dashed).

Table 2. Results of the Wilcoxon signed-rank test comparing reconstruction performance between the CNN (baseline) and KAN-based models across datasets and bottleneck sizes. Tests were paired by dataset and latent dimension. Holm correction was applied to control the family-wise error rate at $\alpha = 0.05$.

Metric	Comparison	n	Statistic	p-Value	p_{Holm}	Significant
MSE	CNN vs FastKAN	20	29.0	0.0032	0.0063	Yes
	CNN vs KAN-linear	20	54.0	0.0583	0.0583	No
PSNR	CNN vs FastKAN	20	19.0	0.0006	0.0012	Yes
	CNN vs KAN-linear	20	40.0	0.0136	0.0136	Yes
SSIM	CNN vs FastKAN	20	48.0	0.0328	0.0655	No
	CNN vs KAN-linear	20	74.0	0.2611	0.2611	No

approaches, preserving structural fidelity while avoiding excessive smoothing and without introducing noticeable artifacts, resulting in more visually consistent reconstructions under strong compression.

5.5. Computational efficiency. Inference time measurements indicate a clear computational advantage for FastKAN. On MNIST and Shapes, FastKAN achieves inference times as low as 0.014–0.031 ms per image, making it suitable for real-time applications. The linear KAN remains efficient on smaller images (e.g., CIFAR-10), with inference times below 0.35 ms even at the largest bottleneck.

The CNN inference time increases substantially with image resolution, reaching approximately 2.6 ms per image on the Large dataset. The order-of-magnitude speedup achieved by FastKAN at high bottleneck widths underscores its practical benefits for deployment scenarios

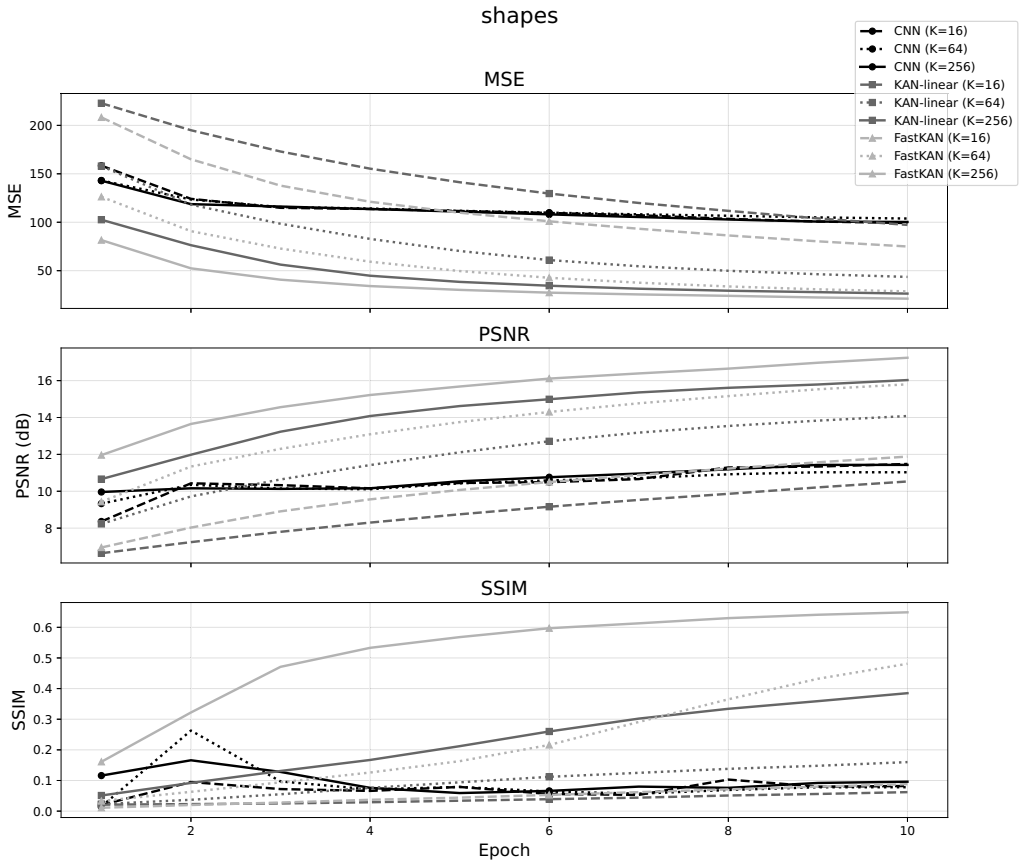

Fig. 5. Training dynamics on the Shapes dataset for selected bottleneck widths K.

that require both high reconstruction quality and low latency.

The reported inference times provide a practical indication of computational performance for the best-performing model within each configuration. However, these results should be interpreted with caution, as they do not correspond to a fully controlled comparison across architectures. In particular, the compared models differ in parameter count, architectural design, and internal function parameterization. Consequently, the observed runtime differences reflect both architectural efficiency and configuration-specific optimization rather than pure intrinsic computational complexity alone.

A fully systematic comparison of inference time would require additional controls, such as matched parameter counts, comparable depth, and standardized hardware utilization across all evaluated models. Although such an analysis is beyond the scope of the present study, it represents an important direction for future work in the rigorous benchmarking of KAN-based and convolutional autoencoders.

Overall, the results demonstrate that KAN-based autoencoders provide a strong alternative to classical convolutional architectures. The linear KAN excels on small natural images, FastKAN consistently delivers the best trade-off between quality and efficiency, and CNNs remain competitive only under extreme compression or limited latent capacity. These findings confirm that function-based representations offer tangible advantages for image reconstruction tasks across a wide range of data regimes.

5.6. Discussion. The results of this study demonstrate that adaptive functional parameterizations in Kolmogorov–Arnold networks enable effective learning of image representations despite the absence of an explicit convolutional structure. Across multiple datasets and compression regimes, KAN-based autoencoders achieve competitive reconstruction quality relative to classical convolutional autoencoders, while exhibiting distinct trade-offs in parameter efficiency and inference speed.

A key observation emerging from the experimental analysis is the strong dependence of model performance on latent dimensionality. For very narrow bottlenecks (e.g., $K = 16$ or 32), convolutional autoencoders often retain a slight advantage, particularly on simple datasets such as MNIST. In these settings, CNNs more effectively preserve the global shape of objects, which

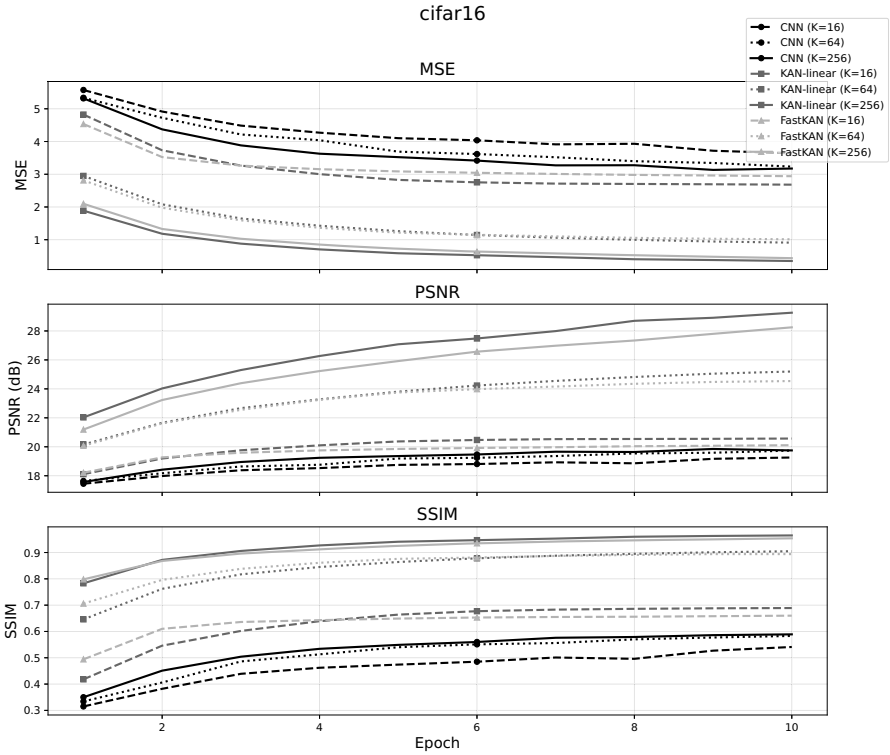

Fig. 6. Training dynamics on the CIFAR dataset for selected bottleneck widths K.

is reflected in higher SSIM values. However, as the bottleneck capacity increases, KAN-based models rapidly close this gap and frequently surpass CNNs in both perceptual quality (SSIM) and signal fidelity (PSNR). This behavior suggests that adaptive univariate function parameterizations are especially effective once sufficient representational capacity is available.

The behavior observed on datasets with sharp geometric structures (Shapes) further highlights the strengths of KAN-based architectures. In this case, FastKAN consistently outperforms CNNs across all bottleneck sizes, achieving substantially higher SSIM values. This indicates that the flexible, edge-wise functional mappings in KANs are particularly well suited to capturing high-frequency structures and preserving contour sharpness, whereas CNNs tend to introduce excessive smoothing under strong compression.

For more complex and higher-resolution natural images (Large dataset), performance differences are more nuanced. CNNs retain a modest advantage at smaller bottleneck sizes, likely due to their strong inductive bias toward local spatial correlations. However, at larger bottleneck widths ($K = 256$), FastKAN becomes competitive or superior, indicating that the representational advantages of adaptive functional mappings emerge more clearly when sufficient latent capacity is available.

Beyond reconstruction quality, computational efficiency constitutes a central contribution of this work. While classical KAN models offer high reconstruction fidelity, they incur a significant computational cost due to the evaluation of spline-based univariate functions. FastKAN addresses this limitation by replacing B-spline parameterizations with radial basis functions, resulting in inference times that are often an order of magnitude lower than those of CNNs on small and medium-scale datasets. This efficiency advantage positions FastKAN as a particularly attractive candidate for real-time and resource-constrained applications.

5.7. Comparison with convolutional autoencoders. The existing research has demonstrated the feasibility of incorporating Kolmogorov–Arnold networks into autoencoder architectures for image representation learning. However, the published results typically emphasize reconstruction accuracy or downstream task performance, and the models are evaluated under differing architectural capacities or without explicit efficiency analysis.

In this paper, we provide a controlled comparison of KAN-based, FastKAN-based, and convolutional autoencoders under strictly matched latent dimensionality constraints. By jointly analyzing reconstruction fidelity (MSE), perceptual quality (SSIM, PSNR), parameter count, and inference time across multiple datasets and bottleneck sizes, this work clarifies the practical trade-offs

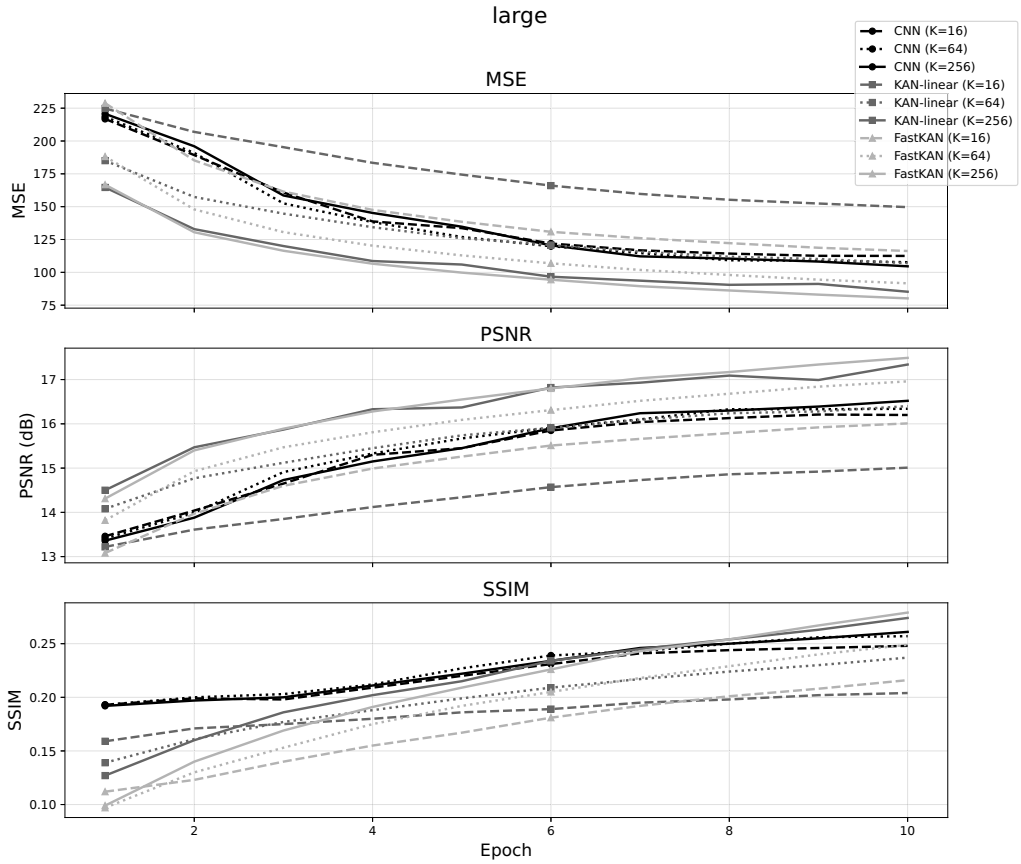

Fig. 7. Training dynamics on the Large dataset for selected bottleneck widths K.

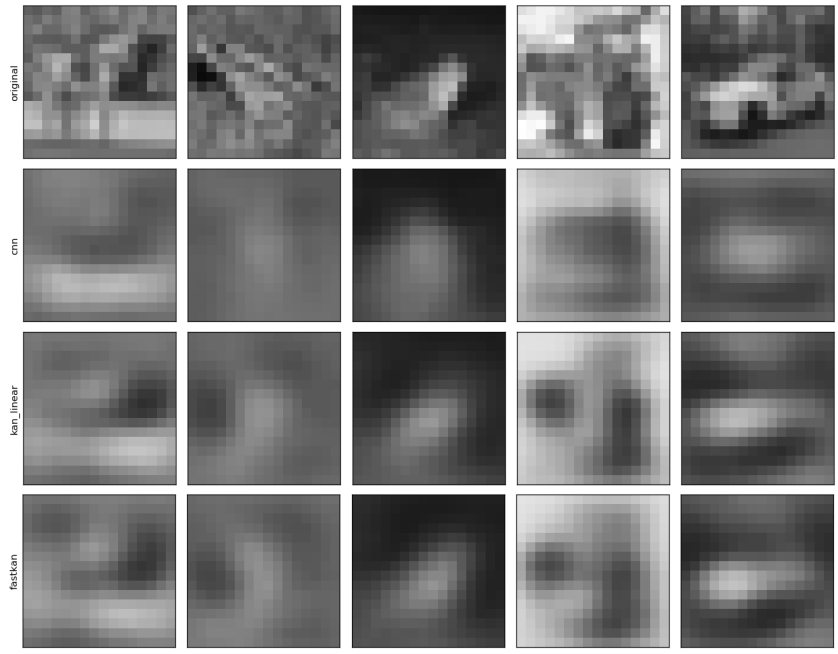

Fig. 8. Qualitative reconstructions on CIFAR-10 (grayscale, 16×16). Top row: original images. Subsequent rows: reconstructions produced by the CNN, KAN, and FastKAN autoencoders for a representative bottleneck setting ($K = 32$).

associated with adaptive functional parameterizations in image autoencoders.

A direct comparison with representative KAN-based image autoencoder studies from the literature is summarized in Table 3. Unlike prior work, this paper explicitly evaluates inference-time efficiency and reports results across both synthetic and natural image datasets under identical training protocols.

The results presented in Table 3 indicate that adaptive functional parameterizations provide a viable and competitive alternative to convolutional structures for image autoencoding. While CNNs remain advantageous in extreme compression regimes and for capturing a coarse global structure, KAN-based models offer greater flexibility and superior performance once adequate latent capacity is available. FastKAN, in particular, achieves a favorable balance between reconstruction quality and computational efficiency, highlighting its practical relevance for deployment-oriented scenarios.

Table 3. Comparison with existing KAN-based image autoencoder studies.

Study	Task	Datasets	Metrics reported	Efficiency analysis
Moradi *et al.*, 2024	Image autoencoder	MNIST, SVHN, CIFAR-10	Reconstruction loss (MSE)	Partial (parameter count, training time)
Yu *et al.* 2025	Representation learning	MNIST, FashionMNIST, CIFAR-10/100	Reconstruction loss, downstream accuracy	No
Results presented in this paper	Image autoencoder	MNIST, CIFAR-10 (16×16), Shapes, STL-10 (64×64)	MSE, PSNR, SSIM	Yes (parameter count and inference time)

6. Conclusions

This work presented a systematic evaluation of Kolmogorov–Arnold network (KAN) architectures for autoencoder-based image representation learning. Unlike conventional convolutional autoencoders, which rely on fixed spatial inductive biases and shared convolutional kernels, KAN-based models employ adaptive univariate function parameterizations that directly reflect the structure of the Kolmogorov–Arnold superposition theorem.

Through a controlled experimental design with strictly matched latent dimensionality, this study demonstrated that KAN- and FastKAN-based autoencoders can achieve reconstruction quality that is competitive with, and in several settings superior to, convolutional autoencoders. Improvements were observed particularly in terms of perceptual fidelity (SSIM) and signal quality (PSNR) when sufficient latent capacity was available. These results indicate that an explicit convolutional structure is not strictly necessary for effective image representation learning when adaptive functional mappings are employed.

A central contribution of this work lies in its efficiency-aware analysis. While spline-based KAN models achieve strong reconstruction performance, they incur higher computational cost due to evaluating learned spline functions on each network edge. The FastKAN variant effectively mitigates this limitation by replacing spline parameterizations with radial basis functions, resulting in substantially reduced inference time while preserving most of the representational benefits of a standard KAN. In several experimental settings, FastKAN achieved a favorable balance between reconstruction accuracy, parameter efficiency, and computational speed,

outperforming convolutional baselines in both perceptual quality and runtime.

In contrast to prior KAN-based autoencoder studies, which primarily focus on reconstruction loss or downstream task performance under varying architectural capacities, this work provided a unified comparison of the KAN, FastKAN, and convolutional autoencoders under identical latent dimensionality and training conditions by jointly analyzing the reconstruction error, perceptual quality, parameter count, and inference time across multiple datasets and bottleneck sizes. The study clarified the practical trade-offs associated with adaptive functional parameterizations in autoencoder architectures.

Overall, the results establish the KAN as a viable and principled alternative to convolutional architectures for image representation learning. The findings suggest that adaptive univariate function modelling offers a complementary inductive bias that can be particularly advantageous when balanced against computational constraints.

Future work. Future research will investigate hybrid architectures that combine convolutional encoders with KAN-based latent mappings to jointly exploit spatial locality and adaptive functional expressiveness. This direction is directly motivated by our empirical findings: CNNs remain competitive at narrow bottlenecks and limited latent capacity, whereas KAN-based models, particularly FastKAN, become more advantageous as latent dimensionality increases. A hybrid design may therefore combine the strong local inductive bias of convolutional structures with the flexible functional modelling of KAN layers.

Extensions to higher-resolution images and multi-channel data are also of interest, as are entropy-aware latent modelling strategies, for example, through coding-aware regularization that explicitly encourages compressible latent representations (in contrast to standard reconstruction-only training), moving the framework closer to explicit neural compression pipelines. Finally, the interpretability of learned univariate functions in KAN layers provides an opportunity to analyze how individual scalar dimensions are transformed across the network. Because these learned mappings can be directly visualized, they provide a more transparent interpretation compared to convolutional models, where nonlinear transformations are distributed across multiple layers and filters.

Acknowledgment

This research is partially supported by the NAWA STARS EU BOOSTER project as part of the FERS.01.05-IP.08-0219/23 programme, under the agreement BPI/WUE/2024/1/00029/U/00001. Joanna Kołodziej's work was additionally supported by the MIRANDA project, which received funding from the European Union's Horizon Europe Research and Innovation Programme under the grant agreement no. 101168144.

Ilona Anna Urbaniak, Sylwester Wieczorek, and Joanna Kołodziej contributed equally to all aspects of this work, including conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing, visualization, supervision, and project administration. All authors have read and agreed to the published version of the manuscript, and declare no conflicts of interest.

MNIST, CIFAR-10, and STL-10 are publicly available datasets. The synthetic Shapes dataset was generated procedurally as described in Section 4.1.

The implementation used in this study will be made available upon publication; until then, it is available from the corresponding author upon reasonable request.

References

Arnold, V.I. (1957). On functions of three variables, *Doklady Akademii Nauk SSSR* **114**(4): 679–681.

Bengio, Y., Courville, A. and Vincent, P. (2013). Representation learning: A review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8): 1798–1828.

Bodner, A.D., Tepsich, A.S., Spolski, J.N. and Pourteau, S. (2024). Convolutional Kolmogorov–Arnold networks, *arXiv* 2406.13155.

Braun, J. and Griebel, M. (2009). On a constructive proof of Kolmogorov's superposition theorem, *Constructive Approximation* **30**(3): 653–675.

Cheon, M. (2024). Kolmogorov–Arnold network for satellite image classification in remote sensing, *arXiv* 2406.00600.

Coates, A., Ng, A.Y. and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning, *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA*, pp. 215–223.

Firsov, N., Myasnikov, E. and Lobanov, V. (2024). HyperKAN: Kolmogorov–Arnold networks make hyperspectral image analysis more accurate, *Sensors* **24**(23): 7683.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*, MIT Press, Cambridge.

Hinton, G.E. and Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks, *Science* **313**(5786): 504–507.

Kolmogorov, A.N. (1957). On the representation of continuous functions of several variables by continuous functions of one variable and addition, *Doklady Akademii Nauk SSSR* **114**(5): 953–956.

Kowalik-Urbaniak, I.A. Castelli, J., Hemmati, N., Koff, D., Smolarski-Koff, N., Vrscay, E.R., Wang, J. and Wang, Z. (2015). Modelling of subjective radiological assessments

with objective image quality measures of brain and body CT images, *in* M. Kamel and A. Campilho (Eds), *Image Analysis and Recognition*, Lecture Notes in Computer Science, Vol. 9164, Springer, Cham, pp. 3–13.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images, *Technical Report TR-2009*, University of Toronto, Toronto, https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE* **86**(11): 2278–2324.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y. and Tegmark, M. (2024). KAN: Kolmogorov–Arnold networks, *arXiv* 2404.19756.

Lobanov, V., Firsov, N., Myasnikov, E., Khabibullin, R. and Nikonorov, A. (2024). HyperKAN: Kolmogorov-Arnold networks make hyperspectral image classificators smarter, *arXiv* 2407.05278.

Lorentz, G.G. (1966). *Approximation of Functions*, Holt, Rinehart and Winston, New York.

Ma, X. Wang, Z., Hu, Y., Zhang, X. and Pun, M.-O. (2025). Kolmogorov–Arnold network for remote sensing image semantic segmentation, *arXiv* 2501.07390.

Moradi, M., Panahi, S., Bollt, E.M. and Lai, Y.-C. (2024). Kolmogorov–Arnold network autoencoders, *arXiv* 2410.02077.

Ouni, A., Samir, C., Bouaziz, Y. and Fradi, A. (2025). ConvKAN: Towards robust, high-performance and interpretable image classification, *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025), Porto, Portugal*, Vol. 2, pp. 48–58.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N. and Antiga, L. (2019). PyTorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems* **32**: 8024–8035.

Penkin, M. and Krylov, A. (2025). FunKAN: Functional Kolmogorov–Arnold network for medical image enhancement and segmentation, *arXiv* 2509.13508.

Pituková, L., Sinčák, P., Kovács, L.J. and Wang, P. (2025). Capsule-ConvKAN: A hybrid neural approach to medical image classification, *arXiv* 2507.06417.

Urbaniak, I.A. and Pinto, R.N. (2023). Quality assessment of medical images, *in* H. Kunze *et al.* (Eds), *Engineering Mathematics and Artificial Intelligence: Foundations, Methods and Applications*, Routledge, Boca Raton, pp. 463–484.

Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P. (2004). Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Processing* **13**(4): 600–612.

Yang, Z., Zhang, J., Luo, X., Lu, Z. and Shen, L. (2025). MedKAN: An advanced Kolmogorov–Arnold network for medical image classification, *arXiv* 2502.18416.

Yeh, C.-H. and Liou, C.-Y. (2025). Diffusion-based low-light image enhancement with Kolmogorov–Arnold networks (KANs), *Array* **27**: 100431.

Yu, F., Hu, R., Lin, Y., Ma, Y., Huang, Z. and Li, W. (2025). KAE: Kolmogorov–Arnold auto-encoder for representation learning, *arXiv* 2501.00420.

Ilona Anna Urbaniak holds a PhD degree in applied mathematics from the University of Waterloo (Ontario, Canada). She is currently an assistant professor in the Department of Computer Science at the Cracow University of Technology (Poland). Her research interests include artificial intelligence, machine learning, and medical image analysis, with a particular focus on representation learning, explainable AI, and deep learning-based image compression. Her work focuses on bridging theoretical foundations and practical applications of machine learning in imaging and diagnostic contexts. She has held research and industry positions in applied AI and data science, contributing to both academic research and real-world applications. ORCID: 0000-0002-1948-6501.

Sylwester Wieczorek is affiliated with the Department of Computer Science at the Cracow University of Technology, Poland. His research interests include artificial intelligence and machine learning, with a focus on deep learning architectures and representation learning for image analysis. His work explores practical applications of neural networks in data driven systems.

Joanna Kołodziej is a professor of computer science at at NASK—National Research Institute in Warsaw. She is the president of the Polish Chapter of the IEEE Computational Intelligence Society. Her research is focused on machine learning and Big Data, security aspects, energy awareness in resource management, data scheduling in clouds, cloud and edge computing, and cybersecurity aspects in ICT systems. ORCID: 0000-0002-5181-8713.

Received: 31 January 2026
Revised: 21 March 2026
Re-revised: 23 March 2026
Accepted: 4 April 2026