VHR Multispectral Satellite Image Classification with Kolmogorov-Arnold Networks for Urban Applications

Mohamed Fawzy[1,2], Arpad Barsi[1]

[1]Department of Photogrammetry and Geoinformatics, Faculty of Civil Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3, H-1111 Budapest, Hungary, {mohamed.fawzy, barsi.arpad}@emk.bme.hu
[2]Civil Engineering Department, Faculty of Engineering, South Valley University, 83523 Qena, Egypt, mohamedfawzy@eng.svu.edu.eg

Keywords: Kolmogorov-Arnold Networks, Multispectral Satellite Imagery, LCLU Mapping, VHR Image Classification.

Abstract

The Kolmogorov-Arnold Network (KAN) is a computational framework rooted in the Kolmogorov-Arnold representation theorem, which states that any continuous function of multiple variables can be expressed as a superposition of univariate functions. KAN leverages this theoretical foundation to decompose complex functions into simpler and lower-dimensional ones. Function decomposition makes KAN particularly suitable for challenging tasks in image classification, where separating data into distinct categories often requires approximating intricate and multi-dimensional boundaries. In practice, KAN, relying on function decomposition, offers an alternative to multilayer perceptron (MLP) architectures. By efficiently encoding nonlinear relationships among features, KAN demonstrates potential in multilayered data analysis tasks such as multi and hyper-spectral remotely sensed image classification. As a result, KAN finds significant applications in remote sensing image processing, particularly in Land Cover and Land Use (LCLU) mapping using Very High-Resolution (VHR) satellite imageries. Due to their fine detail, VHR images provide an exceptional basis for accurately distinguishing between various land cover types. When multispectral data is incorporated, KAN excels by leveraging its ability to model nonlinear relationships, allowing for highly accurate classifications. KAN's performance is validated using ground-truth data collected from field surveys or random reference points visually picked from images. Additionally, KAN is benchmarked against other methods like traditional Shallow Neural Networks (SNNs). Obtained KAN classification accuracy and computational efficiency are evaluated compared to SNN, highlighting its strengths in modelling complexity with optimized key model parameters. A Python-based implementation of KAN is developed for flexible integration into existing geospatial analysis workflows and highlighting its compatibility with cloud computing environments such as Google Colab. This integration enhances scalability, makes it practical for processing large-scale satellite datasets efficiently, and facilitates high-resolution mapping and reproducibility in environmental monitoring and urban applications. The reliability of KAN and the potential classification accuracy of different model architectures were verified. The KAN model with a 10-neuron mid-layer achieved an overall accuracy of 88.89%, outperforming the SNN results with a maximum accuracy of 87.84 for a model with 20 & 20-neuron hidden layers.

1. Introduction

Urban Land Cover and Land Use (LCLU) classification plays a crucial role in remote sensing applications, including environmental monitoring, urban planning, and disaster management. The increasing availability of Very High-Resolution (VHR) multispectral satellite imagery has significantly enhanced the ability to analyse and classify urban environments at a fine spatial scale. However, extracting meaningful information from such high-dimensional data remains challenging due to the complexity of urban landscapes, spectral variability, and computational constraints (Li, Chen et al. 2024). Traditional machine learning approaches, such as Shallow Neural Networks (SNNs) and Convolutional Neural Networks (CNNs), have been widely used for remote sensing classification tasks (Fawzy, Dowajy et al. 2024). While these models demonstrate strong classification capabilities, they often require extensive training data, exhibit high computational complexity, and may suffer from overfitting. Furthermore, deep neural networks lack interpretability, making it difficult to understand how classification decisions are made (Fawzy and Barsi 2024). The traditional shallow and deep neural network challenges highlight the need for alternative computational frameworks that can efficiently model nonlinear spectral relationships while maintaining interpretability and computational efficiency.

Kolmogorov-Arnold Networks (KANs) offer a promising alternative to conventional neural network architectures. Based on the Kolmogorov-Arnold representation theorem, KAN decomposes complex multivariate functions into a series of simpler univariate ones. This unique characteristic allows KAN to approximate high-dimensional functions with fewer parameters, improving generalization and reducing computational cost. Unlike traditional multilayer perceptrons (MLPs), which rely on fixed activation functions, KAN dynamically learns the required functions, providing greater flexibility in modelling spectral variations in multispectral imagery (Liu, Wang et al. 2024).

The main objective of this approach is to investigate the effectiveness of Kolmogorov-Arnold Networks for classifying urban land cover features based on VHR multispectral satellite images. A KAN-based classification framework is developed and evaluated against a traditional SNN to determine its advantages and limitations. The study also investigates how key model parameters influence classification performance, including the number of mid-layers and neurons, grid size, and spline order. The classification results are validated using ground-truth data and statistical accuracy analyses to assess the model's reliability and practical effectiveness.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-245-2025

2. KAN Theoretical Foundation

The Kolmogorov-Arnold representation theorem states that any multivariate continuous function *f*, defined on a bounded domain, can be expressed as a finite composition of continuous univariate functions and additions. Specifically, for a smooth function:

$$f: [0,1]^n \to \mathbb{R} \quad (1)$$

there exists a representation of the form:

$$f(x) = f(x_1, \cdots, x_n) = \sum_{q=1}^{2n+1} \Phi_q\left(\sum_{p=1}^{n} \phi_{q,p}(x_p)\right) \quad (2)$$

where $\phi_{q,p}: [0,1] \to \mathbb{R}$ and $\Phi_q: \mathbb{R} \to \mathbb{R}$ are continuous functions. This decomposition demonstrates that any multivariate function can be constructed using a sum of simpler univariate functions, highlighting an essential property of functional representations. In matrix notation, this theorem can be rewritten as:

$$f(x) = \Phi_{out} \circ \Phi_{in} \circ x \quad (3)$$

Where:

$$\Phi_{in} = \begin{bmatrix} \phi_{1,1}(\cdot) & \cdots & \phi_{1,n}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{2n+1,1}(\cdot) & \cdots & \phi_{2n+1,n}(\cdot) \end{bmatrix} \quad (4)$$

$$\Phi_{out} = [\Phi_1(\cdot) \quad \cdots \quad \Phi_{2n+1}(\cdot)] \quad (5)$$

This formulation illustrates how a complex function can be expressed through layers of simple univariate transformations, forming the conceptual basis for KANs.

Inspired by this theoretical foundation, Kolmogorov-Arnold Networks define a layer where each edge in the network learns a univariate function in the form of an activation function.

A full KAN model consists of multiple layers (L); for an input vector $x_0 \in \mathbb{R}^{n_0}$, the network applies successive transformations such that:

$$x_L = \mathrm{KAN}(x_0) = \Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_0 \circ x_0 \quad (6)$$

In practice, the univariate functions $\phi(x)$ are parameterized as a combination of a smooth activation function (silu) and a B-spline function (spline):

$$\mathrm{silu}(x) = \frac{x}{1+e^{-x}} \quad (7)$$

$$\mathrm{spline}(x) = \sum_i c_i \cdot B_i(x) \quad (8)$$

B-spline basis functions are piecewise polynomial functions. B-splines are numerically stable, offer local support, and allow flexible function approximation, making them well-suited for constructing the functions for layer l at order k $\phi_{l,k}$ and Φ_l. Subsequently, a linear combination of the hidden activations is formed for each class and a SoftMax function is then applied to the logits for obtaining the class probabilities (Jamali, Roy et al. 2024). This design allows KANs to capture complex transformations more efficiently than traditional multilayer perceptrons (MLPs), leveraging the power of the Kolmogorov-Arnold decomposition to improve function approximation capabilities (Igali and Shamoi 2024).

3. Methodology

To meet the research objectives, the presented methodology is applied, and the outcomes are reviewed (Figure 1). The input VHR multispectral image is processed to provide feature vectors with predefined labels. Extracted features are used to train and test the KAN model across varying parameters. The proper trained model, based on the optimal parameters, is selected to classify the full image into the target classes, and the resulting outcomes are assessed considering the overall, user’s and producer’s accuracy. The obtained findings are compared to shallow neural network results for validation and ensuring the model stability.

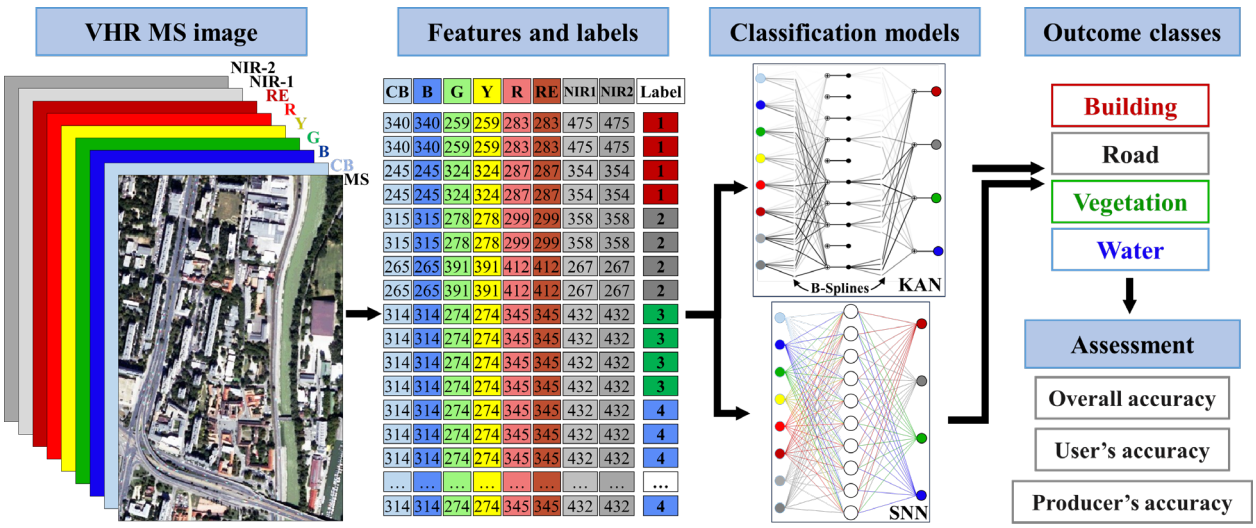

Figure 1. Procedures of the presented methodology.

4. Experimental Works

KAN is a promising solution for multispectral image classification tasks involve image semantic segmentation into discrete classes based on relevant features, which often involve complicated and multi-dimensional limitations. Consequently, KANs offer a wide range of applications in image processing tasks, particularly for mapping urban land cover and land use using Very High-Resolution satellite images (Ma, Wang et al. 2025).

4.1 Study Area and Data Used

Budapest City, the capital of Hungary, is one of Central Europe's historical cities, featuring numerous urban land cover classes such as buildings, roads, vegetation, and water. The spectral characteristics of the urban scenes vary due to the diverse building patterns and materials, the different roof coverings, and the various road and sidewalk pavement materials. The study area is located between latitude 47°32'18.0" & 47°32'54.4"N and longitude 19°02'25.0"E & 19°02'58.0"E, comprises several

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-245-2025 | © Author(s) 2025. CC BY 4.0 License.

building structures, main and secondary roads, a branch of the Danube River, and vegetation areas spread within the entire region (Figure 2).

VHR satellite imageries provide an excellent base for accurately classifying different land cover classes. The used satellite image is acquired by ESA and European Space Imaging through the ‘WorldView-2 European Cities’ project to collect a dataset covering Europe's most densely populated areas between July 2010 and July 2015 (ESA 2025-a). The Worldview-2 satellite, equipped with multispectral sensors, collects data in the coastal blue, blue, green, yellow, red, red-edge, near infrared-1, and near infrared-2 bands included in the employed approach (ESA 2025-b). The spatial resolution is 0.5 m for the panchromatic and 2.00 m for the multispectral bands, therefore the pan-sharpening approach is applied to fuse the panchromatic and multispectral bands, enhancing the raw image for more efficient image analysis tasks. The Principal Component Analysis (PCA) technique is implemented to satellite images where all input bands are pan-sharpened to fit the feature extraction process (Pohl and Van Genderen 2016, Fawzy 2020). The Geodetic Coordinates Reference System (CRS) is World Geodetic System (WGS84) and the projection is Universal Transverse Mercator (UTM) zone 34, EPSG:32634.

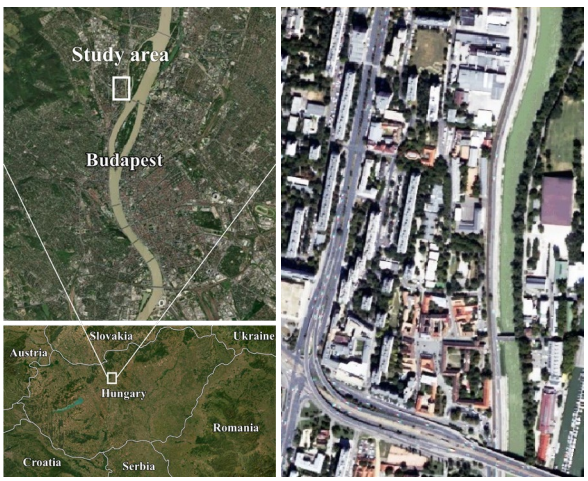

Figure 2. VHR image and study area location, Budapest.

4.2 Feature Vector Extraction

Neural network learning requires significant volumes of features and carefully picked labelled data (Sainos-Vizuett and Lopez-Nava 2021). Handcrafted feature vectors with related labels are extracted from the multispectral image considering pixel-by-pixel spectral value of each band to train and test the classification models. The resulting array includes a set of columns containing the image features and labels, while rows representing pixel values of the flattened image.

4.3 KAN Model

The KAN model relies on numerous parameters including number of mid-layers and neurons, the grid size, and the spline order. Several aspects must be carefully balanced in the practical parameterization of the model. The theoretical framework suggests using 2n+1 neurons in the hidden layer (for example, 17 neurons for n = 8 inputs). However, in practice, altering the number of hidden neurons has significant effects on the model performance. Using fewer neurons may lead to insufficient capacity and underfitting, while an excessive number of neurons can result in overparameterization, causing the model to fit the noise in the training data (overfitting) and adversely affecting generalization. Moreover, increasing the number of neurons in the hidden layer can introduce optimization challenges such as gradient explosion or vanishing gradients (Altarabichi 2024, Firsov, Myasnikov et al. 2024). Although the original Kolmogorov-Arnold formulation employs a single hidden layer, incorporating additional layers can facilitate the learning of hierarchical features; however, deeper networks entail more parameters, further complicate the optimization process, require more hardware resources, and consume extra computational power. Alternative optimizers like SGD with momentum or RMSprop are available with KAN, meanwhile they generally require more careful hyperparameter tuning. The Adam optimizer is often considered as one of the best choices for training KANs due to its adaptive learning rates and robustness to gradient scaling issues, which results in faster and more stable convergence. Even when training on large datasets, comprising tens of thousands of examples, regularization techniques such as L2 weight decay and dropout remain essential to promote generalization and mitigate overfitting. In the context of B-splines, the parameter k determines the order of the polynomial and, therefore, the smoothness of the spline; higher values of k yield smoother approximations but lead to higher computational costs, whereas lower values provide faster but less smooth approximations (Suman, Pacharaney et al. 2024). Although the Kolmogorov-Arnold Network's training process does not include an intrinsic validation step, KAN models can use validation technique as demonstrated in deep learning applications. Validation is necessary for early stopping, which tracks the model's performance during training using the validation dataset. Consequently, training process is stopped if the model's performance on the validation set starts to decrease while it keeps getting better on the training set to prevent the model from overfitting to the training data (Somvanshi, Javed et al. 2024).

4.4 Optimal KAN Model Structure

Identifying the ideal parameters is the most challenging aspect of using KAN since training accuracy varies dramatically based on the number of mid-layers, neurons, grid size, and spline order. The model parameters are adjustable considering the desired classification accuracy level, the targeted application, the computational resources, and the available computing time (Dong, Zheng et al. 2024, Ta, Thai et al. 2025). Multiple trials were conducted to verify the adopted model parameters for finding the optimal balance between training and testing accuracy and time consumption considering available resources. In total, 150 training processes have been applied to adjust the model parameters using one or two mid-layers with neuron numbers N1 and N2 = 10, 20, 30, 40, 50, grid sizes G = 1, 3, 5, and spline order values k = 1, 2 (Figure 3).

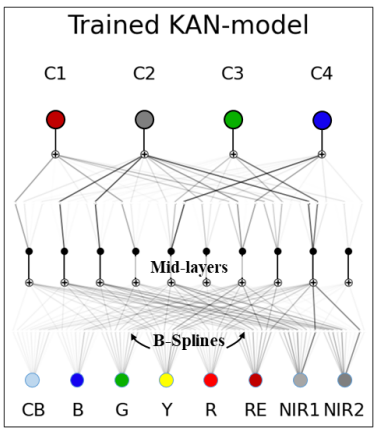

Figure 3. An example of the applied KAN model architecture.

For the prediction phase and image classification, 24 pre-trained models are selected to showcase the KAN performance. The model structures are selected with one or two mid-layers of 10 and 20 neurons, considering the 2n+1 rule, G values of 1, 3, 5 and k values of 1, 2 (Table 1).

Model	Model Parameters				Accuracies (%)		
	N1	N2	G	k	Train	Test	Classification
1	10	-	1	1	89.9	89.7	83.33
2	10	-	1	2	92.9	92.7	81.13
3	10	-	3	1	**96.9**	**96.9**	87.32
4	10	-	3	2	96.2	96.1	85.85
5	10	-	5	1	96.8	96.8	**88.89**
6	10	-	5	2	97.3	97.3	85.74
7	20	-	1	1	80.4	80.0	40.04
8	20	-	1	2	88.8	88.6	50.42
9	20	-	3	1	96.7	96.5	86.06
10	20	-	3	2	95.8	95.7	76.62
11	20	-	5	1	96.7	96.6	77.99
12	20	-	5	2	94.6	94.5	84.07
13	10	10	1	1	50.8	50.5	77.46
14	10	10	1	2	60.8	60.7	77.36
15	10	10	3	1	92.6	92.4	86.69
16	10	10	3	2	94.6	94.5	81.55
17	10	10	5	1	92.7	92.5	84.07
18	10	10	5	2	90	89.6	76.83
19	20	20	1	1	63.3	63.3	55.14
20	20	20	1	2	52.1	51.9	53.98
21	20	20	3	1	94.6	94.5	73.79
22	20	20	3	2	95.7	95.4	80.40
23	20	20	5	1	91.6	91.7	86.27
24	20	20	5	2	94.9	94.6	86.27

Table 1. KAN model parameters and accuracies.

4.5 KAN Model Prediction

Based on the pre-defined features and the target classes, the trained models were applied to classify the entire scene. The resulting classified images (Figure 4) were visually evaluated to explore the effectiveness of the KAN models for image classification across different classes.

For most models, training and test accuracy are similar (e.g., Model 6 with 97.3/ 97.3 %), suggesting that the test set is drawn from a distribution similar to the training set, and overfitting within that split is limited. However, the classification accuracy (e.g., 85.74 %) can diverge substantially from the test accuracy, implying that the final large-scale feature of the classified image differs enough in its distribution to expose gaps in generalization. Some models show large drops (e.g., Model 7 with 80.4/ 80.0 & 40.04 %), indicating significant mismatch or overfitting, what worked on the training/ test split does not carry over well to the broader imagery. Meanwhile, in a few cases (e.g., Model 13), classification accuracy (77.46 %) exceeds the training/ test accuracy (50.8/ 50.5 %), suggesting that the training set may have been less representative of the complete set of labelled points, or the larger dataset happens to align better with the learned decision boundaries.

Models 1 – 12 are single-mid-layer KANs; among these, increasing neuron numbers from 10 to 20 does not improve training/ test scores (e.g., Model 3 vs. Model 9), so adding more neurons alone does not guarantee higher classification accuracy on the full dataset. Model 5 (N1=10, G=5, k=1) notably reaches the highest classification overall accuracies (88.89 %), outperforming networks with more neurons. Models 13 – 24 employ double mid-layer KANs; in principle, this higher capacity can learn more nuanced patterns. For example, Model 15 (N1=10, N2=10, G=3, k=1) and Model 23 (N1=20, N2=20, G=5, k=1) achieve classification accuracies above 86 %. Still, the top two-layer results (e.g., Model 15 at 86.69 %) do not surpass the best single-layer model (Model 5 at 88.89 %), implying that the additional layer may lead to diminishing returns or even instability unless carefully tuned and backed by sufficient training data.

Size of grid G = 1 with more than one mid-layer causes large misclassifications where some classes were totally missed. Meanwhile, models with G = 3 or 5 generally show higher train/ test accuracies than those with G = 1. For instance, compare Model 1 (G = 1) vs. Model 3 (G = 3) or Model 5 (G = 5): in each case, increasing G tends to raise both training and test scores.

In single-layer networks (Models 1 – 6), going from G = 3 to G = 5 can improve classification accuracy (e.g., Model 3 at 87.32 % vs. Model 5 at 88.89 %). However, unfortunately, if other parameters are misaligned, higher G can yield unpredictable drops (e.g., Model 10 with 95.7 % test, but 76.62 % classification). A lower spline order (k = 1) typically generates smoother boundaries, sometimes leading to fewer scattered misclassifications. For instance, Model 5 (k = 1) achieves 88.89 % classification, notably higher than Model 6 (k = 2) at 85.74 %. In some configurations (N = 20 and G = 5), a higher k can be beneficial or at least not harmful (compare Models 23 and 24, both at 86.27 % classification). Hence, the impact of k is strongly tied to G and N; no universal advantage was found to always choosing k = 1 or k = 2. The best single-layer performance is seen in Model 5 (N1 = 10, G = 5, k = 1) and the best two-layer model is Model 15 (N1 = 10, N2 = 10, G = 3, k = 1), reaching 86.69 %. A balanced tuning of G, k, and N is critical. Higher G (3, 5) often enhances classification accuracy by allowing finer spectral discrimination, but this benefit depends on matching the network complexity (through N and k) to the training data. The key is to balance capacity and generalization so that the network's predictive ability on unseen, real-world examples remain robust.

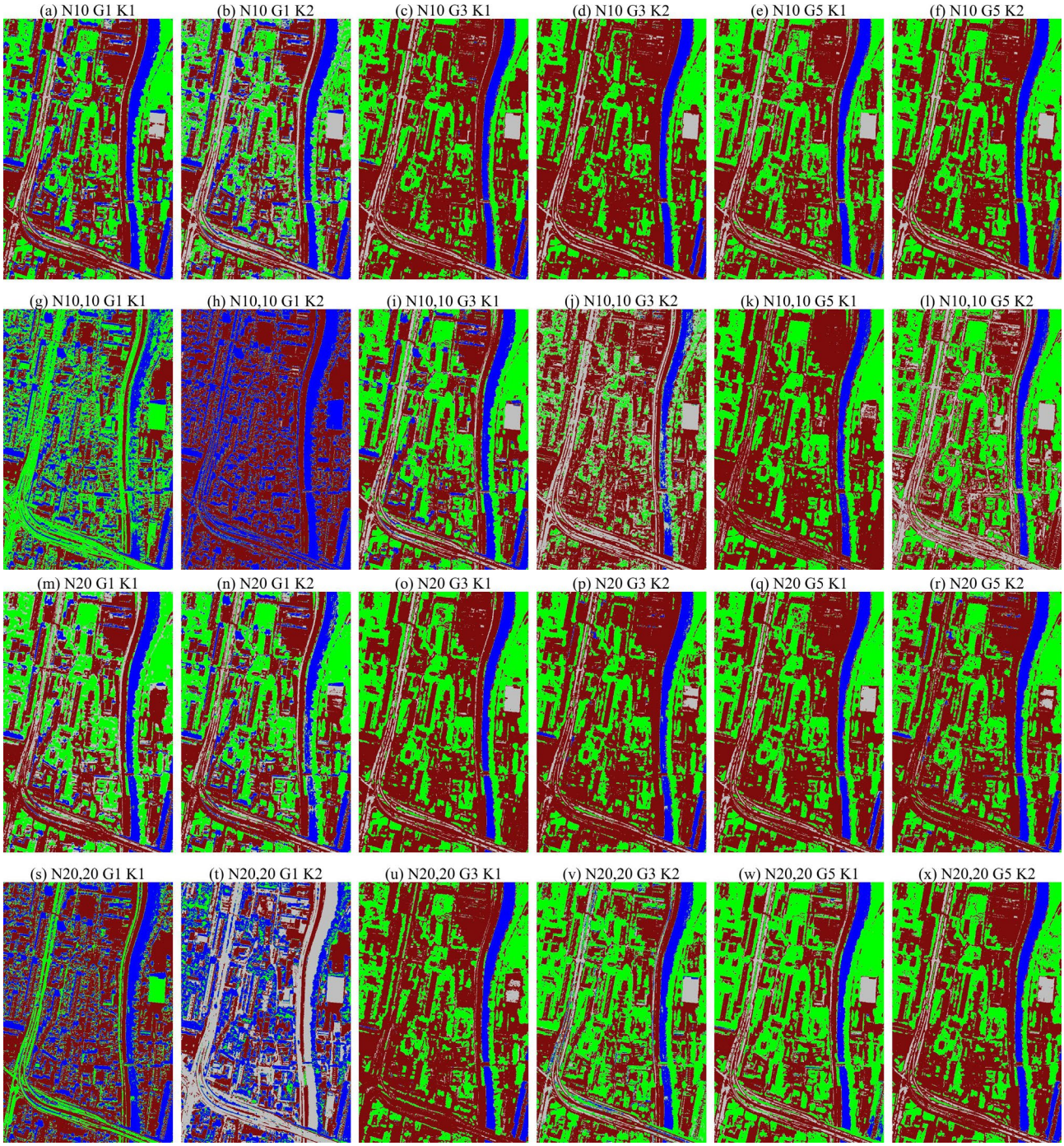

Figure 4. Classification outcomes using different KAN models.

4.6 SNN Model

The shallow neural network parameters should be carefully adjusted to maximize classification findings (Lei, Liu et al. 2020). The SNN model focuses on optimizing input features, number of layers and neurons, and processing time of the network considering the required classification level for a particular application (Fawzy, Szabó et al. 2023). To achieve rigorous outcomes using a SNN, a model with an identical architecture was designed, and applied using the same dataset. For comparison purposes, the SNN model architecture was aligned with the structure of the KAN model that yielded the optimal training, testing and classification accuracy. Four SNN models with one or two mid-layers of 10, 20 neurons were trained, tested, and validated for image classification (Table 2), and their outcomes have been presented (Figure 5).

Model	Model Parameters		Accuracies (%)			
	N1	N2	Train	Test	Validation	Classification
1	10	-	97.7	97.6	97.7	87.32
2	20	-	97.7	97.5	97.7	87.84
3	10	10	97.9	97.9	97.8	87.21
4	20	20	**98.3**	**98.0**	**98.2**	**87.84**

Table 2. SNN model parameters and accuracies.

https://doi.org/10.5194/isprs-annals-X-G-2025-245-2025

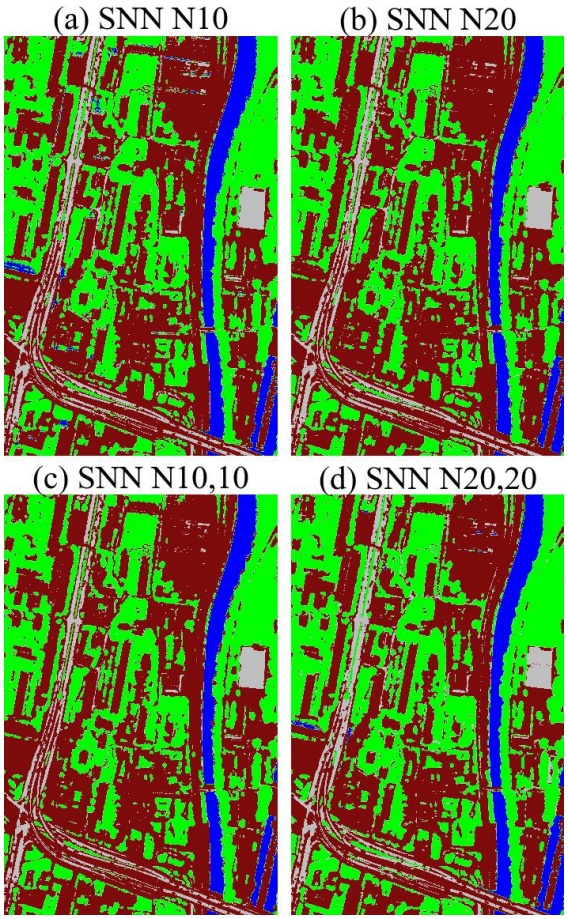

Figure 5. Classification outcomes using different SNN models.

The qualitative evaluations revealed that KAN achieve promising image classification results that are consistent with SNN outcomes; meanwhile, a quantitative assessment is required for a better comprehension of the findings when compared to real-world observations.

5. Results and Discussion

5.1 Accuracy Assessment

To assess quantitative findings, a confusion matrix is applied to figure out the probability that each pixel in the classified image (column values) meets the real-world land cover class (row values) through overall accuracy (Eq. 9), user's accuracy (Eq. 10), and producer's accuracy (Eq.11):

$$\text{Overall Accuracy} = (TP+TN)/(TP+TN+FP+FN) \quad (9)$$

$$\text{Producer's Accuracies} = TP/(TP+FN) \quad (10)$$

$$\text{User's Accuracies} = TP/(TP+FP) \quad (11)$$

True Positive (TP) indicates the number of class pixels that precisely meet the real-world ones, False Positive (FP) refers to the number of non-class pixels identified as class ones, True Negative (TN) represents the number of non-class pixels correctly categorized, and False Negative (FN) denotes the number of class pixels incorrectly classified as non-class ones (Congalton 1991). Confusion matrices are produced for the outcomes of KAN and SNN models using 954 reference points dispersed randomly across all classes with minimum of 105 points per class, and accuracies are graphically presented (Figure 6, 7, 8). Tables 3 and 4 demonstrate the optimum accuracy of the KAN and SNN models.

Classified Data	Building	Road	Vegetation	Water	Total Row	Producer's Accuracy (%)	User's Accuracy (%)
Building	420	24	6	0	450	93.33	85.02
Road	66	182	2	0	250	72.80	88.35
Vegetation	4	0	145	0	149	97.32	94.77
Water	4	0	0	101	105	96.19	100.00
Total column	494	206	153	101	954	-	-
Overall accuracy **88.89%**							

Table 3. Confusion matrix of KAN model (N10, G5, k1).

Classified Data	Building	Road	Vegetation	Water	Total Row	Producer's Accuracy (%)	User's Accuracy (%)
Building	429	17	4	0	450	95.33	82.34
Road	92	156	2	0	250	62.40	89.66
Vegetation	0	1	148	0	149	99.33	96.10
Water	0	0	0	105	105	100.00	100.00
Total column	521	174	154	105	956	-	-
Overall accuracy **87.84%**							

Table 4. Confusion matrix of SNN model (N20, 20).

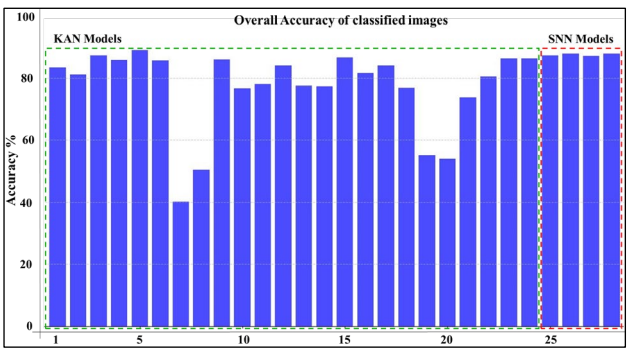

Figure 6. Overall accuracy of the KAN and SNN models.

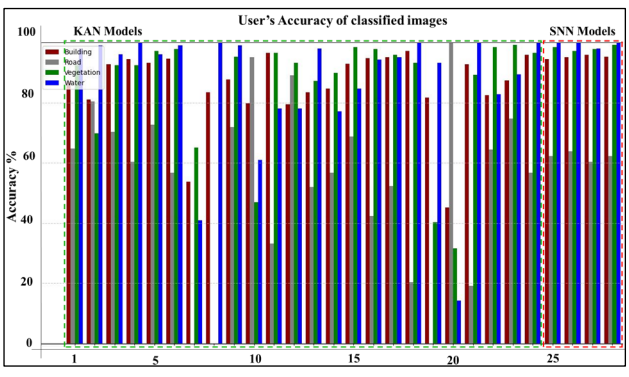

Figure 7. User's accuracy of the KAN and SNN models.

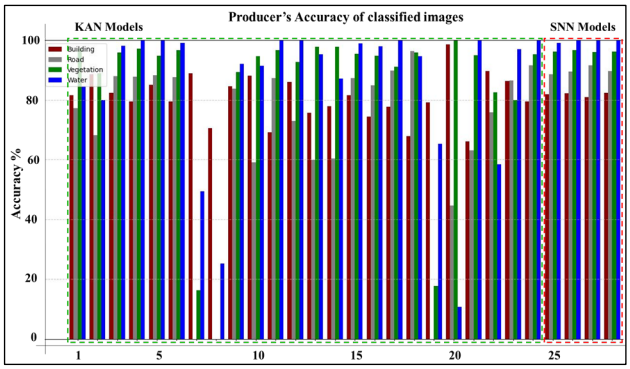

Figure 8. Producer’s accuracy of the KAN and SNN models.

6. Discussions

Analysis of the obtained findings and statistics reveals that increasing the number of neurons (e.g., from N10 to N20) generally provides the Kolmogorov-Arnold Network with greater capacity to learn complex patterns from the input data, allowing it to capture finer class boundaries. For example, when using a single hidden layer with N10, G3, k1, the model can distinguish broad transitions between vegetation and built-up areas, but switching to N20, G3, k1 tends to sharpen these boundaries and reduce confusion near urban edges. However, such heightened capacity also increases the risk of overfitting. In two-layer configurations (N10, 10 or N20, 20), the network can approximate even more intricate data features yet might introduce “salt-and-pepper” noise in otherwise uniform regions if the parameters are not carefully calibrated (e.g., N10, G1, K1 & N10, 10, G1, K1).

The grid parameter G determines the granularity with which the KAN partitions the input space, influencing the level of detail in the classification map. At G = 1, the classification often yields numerous erroneously scattered water pixels across vegetated or urban areas, indicating misclassification caused by the coarse graining. Conversely, higher values such as G = 3 or G = 5 increase the network’s sensitivity to subtle variations. For instance, comparing N10 G1 k2 versus N10 G5 k2 typically reveals that the G = 5 setting yields crisper boundaries around roads and built-up structures. However, it may also exaggerate minor spectral differences, creating fragmented patches in regions that should ideally remain homogeneous, such as continuous vegetation covers.

The parameter k indicates the polynomial degree of the B-splines utilized in the Kolmogorov-Arnold framework. A lower spline order (e.g., k = 1) often produces smoother decision surfaces (i.e., it means simpler spectral decision boundaries), which can be beneficial when focusing on broad land-cover distinctions. For example, a configuration of N10 G3 k1 may generate cohesive clusters of vegetative areas with fewer small-pixel artifacts. On the other hand, higher spline orders (e.g., k = 2 or 3) allow the network to fit more complex local curvature, potentially distinguishing minor spectral differences in transitional zones, such as the interface between water bodies and “shoreline” vegetation. Nevertheless, this added flexibility can also induce over-segmentation in large, uniform regions if not supported by sufficient training data.

The most accurate and robust classification emerges when neuron number N, graining sensitivity G, and spline order k are jointly tuned to match the complexity of the imagery. For instance, N20, 20, G5, k2 might excel in identifying subtle urban features and narrow roads while maintaining sharp vegetation boundaries, yet it can sometimes produce scattered misclassifications in uniform water surfaces. Conversely, a more moderate setting, such as N10, G3, k1, may yield cleaner, less noisy results but might overlook subtle transitions. Hence, finding the right balance among these parameters is crucial for distinguishing vegetation, built-up areas, roads, and water bodies accurately, while minimizing speckling and other forms of overfitting.

7. Conclusions and Future Works

The study investigated the effectiveness of Kolmogorov Arnold Networks for very high-resolution satellite image classifying in urban land cover and land use mapping. The performance of KAN was compared with shallow neural networks to assess the classification accuracy, the impact of different network parameters, and generalization capabilities. Results showed that network architecture significantly influences classification performance. While increasing the number of hidden layers and neurons can improve learning capacity, it does not always translate into better classification results. The best-performing model was a single-layer KAN (Model 5) achieving 88.89% overall classification accuracy, outperforming even the best two-layer KAN models. This suggests that a more complex architecture does not necessarily yield better outcomes and may even lead to instability if not properly optimized.

Among the parameters tested, grid size G and spline order k had the greatest impact on accuracy. Larger grid sizes (G = 3, G = 5) generally improved classification accuracy by capturing finer spectral details. However, overly fine partitioning also increased sensitivity to spectral noise. The spline order affected decision boundaries, with k = 1 producing smoother classifications and k = 2 capturing more complex variations but sometimes introducing unnecessary segmentation. Overfitting was a notable issue in some cases, particularly when models were complex relative to the dataset size. Certain models exhibited high training/ test accuracy but significantly lower classification accuracy, indicating weak generalization. For instance, Model 6 achieved 97.3% test accuracy but only 85.74% classification accuracy, highlighting the risk of tuning models too aggressively to training data. Some cases, like Model 7 (80.4% test accuracy vs. 40.04% classification accuracy), showed extreme overfitting, suggesting that careful balance between model complexity and dataset representativeness is required. Balanced tuning of G, N, and k is essential. Higher G (3, 5) regularly allows for better spectral discrimination, which increases classification accuracy, although the effect depends on matching the network complexity (by N and k) to the training data. A balance between capacity and generalisation is crucial for the network to preserve the strong prediction capabilities on unseen, real-world samples.

The comparison between KAN and SNN models revealed that KAN performed slightly better, with its best model reaching 88.89% classification accuracy, compared to 87.84% for the best SNN model. While the difference was modest, KAN demonstrated advantages in handling nonlinear relationships with fewer parameters, which is particularly useful for high-dimensional satellite image classification. The obtained classification maps showed that KAN produced more precise separations between land cover classes, especially in vegetation and road classification. However, SNNs performed slightly better in distinguishing building classes, indicating that each approach has strengths depending on the specific classification task.

Future work should focus on further optimizing KAN architectures by exploring deeper networks and integrating validating as well as regularization techniques to improve generalization. Combining KAN with CNN-based feature extraction could enhance classification performance, particularly for distinguishing fine-scale urban features. An expansion to

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-245-2025 | © Author(s) 2025. CC BY 4.0 License.

include larger and multi-temporal established databases, such as Copernicus Global Land Cover, Global Land Cover and Land Use Change, or Corine Land Cover, would help to validate KAN's robustness in different real-world scenarios. Additionally, leveraging cloud computing environments could improve the scalability of KAN for large-scale remote sensing applications.

KAN has shown strong potential for urban land classification, offering a competitive alternative to traditional neural networks. However, achieving optimal performance requires careful architecture design and parameter tuning to ensure accurate and generalizable results.

Acknowledgements

The research reported in this paper is part of project no. BME-NVA-02, implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021 funding scheme.

References

Altarabichi, M. G., 2024: Rethinking the function of neurons in KANs. arXiv preprint arXiv:2407.20667.

Congalton, R. G., 1991: A review of assessing the accuracy of classifications of remotely sensed data. Remote sensing of environment 37(1): 35-46.

Dong, C., L. Zheng and W. Chen 2024: Kolmogorov-arnold networks (KAN) for time series classification and robust analysis. International Conference on Advanced Data Mining and Applications, Springer.

ESA, 2025-a: WorldView-2 European Cities: Retrieved 08-Febrauray, 2025, from https://earth.esa.int/eogateway/catalog/worldview-2-european-cities.

ESA, 2025-b: WorldView-2 Instruments." WorldView Series Retrieved 08-Febrauray, 2025, from https://earth.esa.int/eogateway/missions/worldview-2.

Fawzy, M., 2020: Urban Feature Extraction from High Resolution Satellite Images. Master thesis, Faculty of Civil Engineering, South Valley University, Egypt.

Fawzy, M. and A. Barsi, 2024: A U-Net Model for Urban Land Cover Classification Using VHR Satellite Images. Periodica Polytechnica Civil Engineering.

Fawzy, M., M. Dowajy, T. Lovas and A. Barsi, 2024: Urban Land Cover Classification Using Deep Neural Networks Based on VHR Multi-Spectral Image and Point Cloud Integration. IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium, IEEE.

Fawzy, M., G. Szabó and A. Barsi, 2023: A Shallow Neural Network Model for Urban Land Cover Classification Using VHR Satellite Image Features. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 10: 57-64.

Firsov, N., E. Myasnikov, V. Lobanov, R. Khabibullin, N. Kazanskiy, S. Khonina, M. A. Butt and A. Nikonorov, 2024: HyperKAN: Kolmogorov–Arnold Networks Make Hyperspectral Image Classifiers Smarter. Sensors (Basel, Switzerland) 24(23): 7683.

Igali, A. and P. Shamoi, 2024: Image Classification using Fuzzy Pooling in Convolutional Kolmogorov-Arnold Networks. 2024 Joint 13th International Conference on Soft Computing and Intelligent Systems and 25th International Symposium on Advanced Intelligent Systems (SCIS&ISIS), IEEE.

Jamali, A., S. K. Roy, D. Hong, B. Lu and P. Ghamisi, 2024: How to learn more? Exploring Kolmogorov–Arnold networks for hyperspectral image classification. Remote Sensing 16(21): 4015.

Lei, F., X. Liu, Q. Dai and B. W.-K. Ling, 2020: Shallow convolutional neural network for image classification. SN Applied Sciences 2(1): 97.

Li, Z., B. Chen, S. Wu, M. Su, J. M. Chen and B. Xu, 2024: Deep learning for urban land use category classification: A review and experimental assessment. Remote Sensing of Environment 311: 114290.

Liu, Z., Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou and M. Tegmark, 2024: Kan: Kolmogorov-Arnold networks. arXiv preprint arXiv:2404.19756.

Ma, X., Z. Wang, Y. Hu, X. Zhang and M.-O. Pun, 2025: Kolmogorov-Arnold Network for Remote Sensing Image Semantic Segmentation. arXiv preprint arXiv:2501.07390.

Pohl, C. and J. Van Genderen, 2016: Remote sensing image fusion: A practical guide, Crc Press.

Sainos-Vizuett, M. and I. H. Lopez-Nava, 2021: Satellite Imagery Classification Using Shallow and Deep Learning Approaches. Mexican Conference on Pattern Recognition, Springer.

Somvanshi, S., S. A. Javed, M. M. Islam, D. Pandit and S. Das, 2024: A survey on Kolmogorov-Arnold Network. arXiv preprint arXiv:2411.06078.

Suman, S., U. Pacharaney, V. K. Choudhary, R. Raushan, A. Kumar and G. Kumar, 2024: Kolmogorov Arnold Network for Hyperspectral Image Classification: A Detailed Explanation. 2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI), IEEE.

Ta, H.-T., D.-Q. Thai, A. Tran, G. Sidorov and A. Gelbukh, 2025: PRKAN: Parameter-Reduced Kolmogorov-Arnold Networks. arXiv preprint arXiv:2501.07032.