Vol. 7 No. 1, 2026, pp. 1-15 **Jurnal Pepadun**

Evaluating Kolmogorov-Arnold Networks for Multispectral Land Cover Classification Using Sentinel-2 Imagery in Jambi City

[*1]Akhiyar Waladi and [2]Hasanatul Iftitah

[1,2] Faculty of Science and Technology, Universitas Jambi, Jl. Jambi-Muara Bulian KM. 15, Jambi, Indonesia
e-mail: [*1]**akhiyar.waladi@unja.ac.id**, [2]hasanatul.iftitah@unja.ac.id

Abstract *- Land cover maps obtained from satellite imagery are used in environmental management and spatial planning. Deep learning now outperforms traditional machine learning for this task, but Kolmogorov-Arnold Networks (KAN) have rarely been tested on multispectral remote sensing data. This paper evaluates two KAN strategies for classifying nine land cover types from Sentinel-2 imagery in Jambi, Indonesia. ResNet-KAN adds a KAN-based classifier head to a standard CNN backbone, while ConvKAN builds the entire network from KAN-based convolution layers. Both are compared against seven CNN, Transformer, and machine learning baselines using 23 spectral features with Google Dynamic World labels as reference, and ablation experiments test spectral feature composition, ImageNet transfer learning, and input patch size. Swin Transformer reaches the highest overall accuracy (88.34%), but ConvKAN better separates rare land cover classes like Grass and Shrub, achieving the best F1-Macro (0.5870) with only 2.91 million parameters, 89.4% fewer than Swin-T. Adding spectral indices raises ConvKAN F1-Macro by 13.8%, but lowers ViT accuracy by 3.19% OA because self-attention can already learn band-ratio operations from raw bands. KAN models also perform better when trained from scratch, because most Sentinel-2 channels fall outside the visible spectrum that ImageNet covers. Spatially, ConvKAN produces maps as clean as Swin Transformer despite being ten times smaller. KAN can therefore match larger models in accuracy and map quality for multispectral land cover classification.*

Keywords**:** *Kolmogorov-Arnold Networks; Convolutional KAN; Land Cover Classification; Sentinel-2; Remote Sensing.*

1. INTRODUCTION

Forest conservation, agricultural planning, and urban development all depend on accurate information about what covers the land surface and how that coverage shifts year to year [1]. Multispectral satellites now allow repeated mapping over large areas [2]. Sentinel-2 [2] captures ten spectral bands, from visible blue (490 nm) to shortwave infrared (2190 nm), at 10–20 m pixels, revisits every five days, and is freely available. Land use in tropical regions can change within a single growing season. Sentinel-2 revisits frequently enough to capture seasonal changes, but the raw spectral bands cannot always distinguish one land cover type from another. Irrigated cropland and dry grassland reflect sunlight almost identically across all ten bands. NDVI and NDWI values can address this issue. [3]. NDVI picks up the vegetation signal. NDWI detects water. Once computed, these ratios separate the two covers that raw bands values cannot.

Ensemble tree methods such as Random Forest [4] and gradient boosting [5] remain popular for pixel-level land cover mapping because both trains fast and need minimal hyperparameter tuning. Random Forest copes well with many input features, tolerates correlated predictors, and reports variable importance as a training by-product [4]. A shared limitation of all pixel-based classifiers is that each pixel receives a label independently, with no information about its neighbors [6], [7]. Tree plantations and natural forests, for example, reflect almost identically across all ten bands. The only way to tell them apart is their spatial arrangement, since plantation trees are planted in uniform grids while natural forest canopies are randomly distributed. Deep learning works around this problem by taking image patches as input instead of single pixels, so each prediction uses both spectral values and the spatial layout of surrounding pixels. Tens to hundreds of convolutional layers can be trained so the training signal doesn't fade away as it travels back through many layers using Skip Connections in ResNet [8]. The first few layers respond to edges and spectral transitions. Subsequent layers combine these detections into broader spatial patterns such as canopy boundaries [9]. Vision Transformers [10], [11], [12] operate on a different principle and attend to all spatial positions within a patch at once, so a

© 2026 The Authors | doi: 10.23960/pepadun.v7i3.334

pixel at one corner of the patch can influence the label of a pixel at the opposite corner [13]. Models initialized with ImageNet-pretrained weights tend to outperform randomly initialized ones. The pretrained filters have already learned low-level visual features such as edge detectors and texture patterns from millions of natural images [14]. These models still have drawbacks, however. GPU hardware is needed for both training and inference, and such hardware is not always accessible in operational mapping settings [15], [16], [17]. Fixed activation functions like ReLU also apply the same linear response to each input, so they cannot separate two classes (e.g., oil palm and natural forest) whose reflectance values overlap across most bands.

Liu et al. (2024) [18] recently introduced Kolmogorov-Arnold Networks (KAN), a network design that replaces fixed activation functions at each neuron with learnable activation functions at each edge connection. This change is motivated by the Kolmogorov-Arnold theorem, which shows that any continuous multivariate function can be decomposed into two layers of single-variable functions and addition. In practice, KAN realizes these edge functions as B-spline curves, which are piecewise polynomials that can be reshaped during training to fit the specific spectral characteristics of the data while using fewer parameters than a standard MLP of the same depth. For multispectral land cover classification, this matters because overlapping spectral signatures (such as oil palm with NDVI 0.6-0.8 and secondary forest with NDVI 0.7-0.9) cannot be separated by the single-slope response of ReLU. B-spline activations, by contrast, learn different response curves for overlapping input ranges, effectively carving out nonlinear decision boundaries where fixed activations would fail. Building on this idea, convolutional extensions (ConvKAN) replace standard convolution filters with KAN-based spline functions [19], and recent work has demonstrated KAN's effectiveness in scene classification [20] and hyperspectral image classification [21], [22].

We hypothesize that KAN can be a replacement for MLP in multispectral land cover classification with improved accuracy. First, the Kolmogorov-Arnold theorem provides theoretical assurance that a complex mapping from a 23-dimensional spectral space to land cover classes can be decomposed into a learnable univariate function. The B-spline activation function can approximate more flexibly than a fixed ReLU. Second, previous experiments from satellite image classification show that KAN achieves good accuracy with significantly fewer parameters. Cheon [20] showed that KAN hidden layer nodes can be reduced from 256 to 32 without loss of accuracy. Wang et al. (2026) [23] reported Kappa improvements of up to 0.3840 (Bay Area dataset) and 0.2122 (River dataset) compared to MLP on five hyperspectral datasets. Third, a systematic comparison between B-spline and ReLU activations reveals error rate reduction of up to 51% in image classification, with the B-spline model achieving a final error rate a ReLU model cannot match regardless of capacity or training duration increases. These findings suggest that KAN-based architectures should be systematically evaluated against CNN and Transformer baseline models for multispectral land cover mapping.

However, no study has tested KAN for multispectral land cover mapping with systematic ablation experiments. A tropical test site like Jambi, Indonesia, is a suitable proving ground, allowing KAN's performance to be evaluated under a wide variety of land cover conditions. The region presents several classification challenges. Oil palm and rubber plantations share the same NDVI range as natural forest, mixed agricultural plots blend into neighboring land cover without clear borders, and rapid land use change requires frequent remapping [24], [25]. We designed a series of stepwise experiments using two KAN variants trained on Sentinel-2 imagery of Jambi Province, as there is very little previous research applying KAN to land cover mapping with multispectral data. ResNet-KAN retains the pre-trained CNN backbone architecture and only replaces the classification layer at the end with a B-spline layer. ConvKAN completely abandons standard convolution and builds each hidden layer from a B-spline function. ConvKAN abandons standard convolutions entirely and builds every layer from B-spline functions. Seven baselines across CNN, Transformer, and ML families classify the same nine Dynamic World land cover classes. Three ablation experiments then measure the effect of spectral indices, ImageNet pretraining, and patch size on each architecture. ConvKAN produces the most balanced predictions across all classes while using far fewer parameters than the largest model in the study.

2. RESEARCH METHODOLOGY

The methodology follows a five-stage pipeline from data acquisition to model evaluation. Figure 1 shows the data flow of Sentinel-2 image acquisition and Dynamic World label extraction including downscaling and

 | doi: 10.23960/pepadun.v7i3.334

filtering. The process continues from the spectral pre-processing and data patch construction stages, model training with four architecture families, and evaluation through ablation experiments using several metrices.

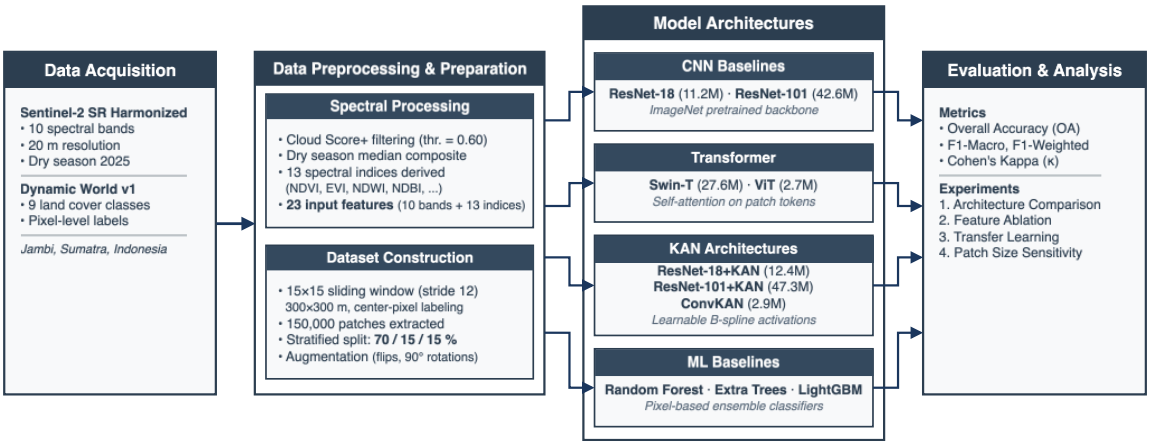

Figure 1. Methodology pipeline: (1) data acquisition, (2) preprocessing, filtering, spectral index derivation, (3) patch extraction and splitting, (4) model training across 4 families, and (5) evaluation with ablation studies.

2.1. Study Area

Jambi Province is located in eastern Sumatra as shown in Figure 2, at the geographic coordinates 1°20'–1°45'S and 103°30'–104°00'E with a total area of approximately 53,435 km². Rainfall ranges between 2200 and 2800 mm per year with distinct rainy and dry seasons. Twenty years ago, the province was largely covered by lowland rainforest and peat swamps. Since then, much of this forest has been cleared for oil palm and rubber plantations [25]. Rice fields, scrubland, and urban development around Jambi City fill the non-plantation areas. The Batang Hari River crosses the province. A problem encountered in satellite image classification is that oil palms often appear almost identical to natural forests in Sentinel-2 imagery. NDVI for both is between 0.7 and 0.9. Red-edge bands do not help much either. Rubber is a different kind of problem. The trees lose all their leaves during dry months, so their NDVI can drop from above 0.7 to below 0.4. A rubber field photographed in July will look completely different from the same field in January, and on smallholder farms where rubber is planted next to fruit trees, there is no clear line between one land cover and the next.

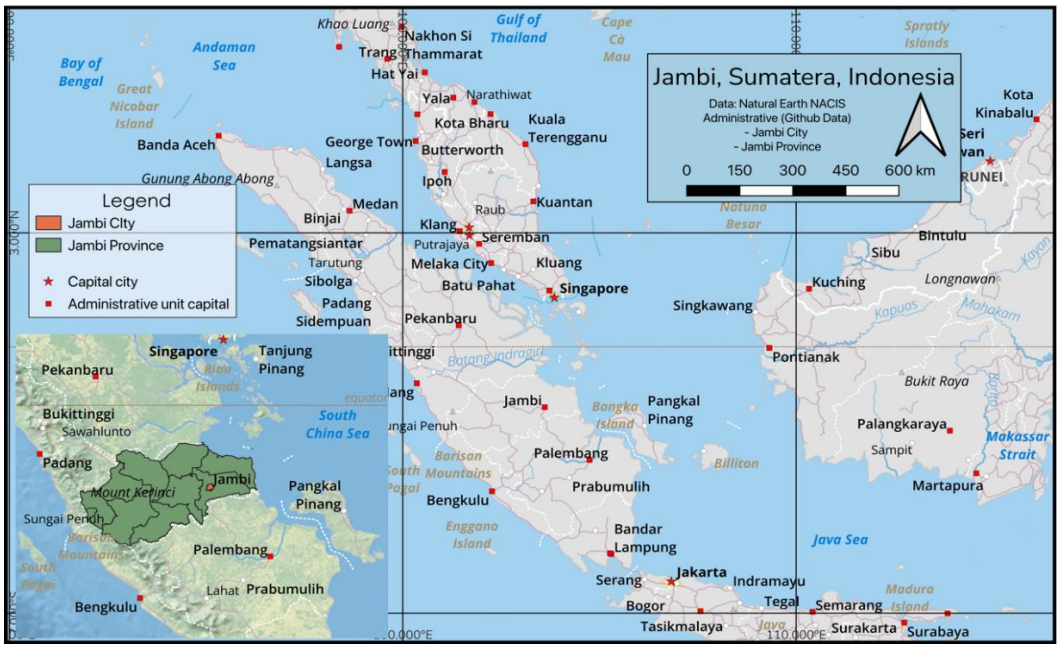

Figure 2. Location of the study area. The main map shows Jambi Province (green) in Sumatra, Indonesia. The inset highlights Jambi City and surrounding districts within the province.

 | doi: 10.23960/pepadun.v7i3.334

2.2. Data Acquisition and Preprocessing

We downloaded Sentinel-2 scenes that had already been atmospherically corrected to surface reflectance (Level-2A product) through the Google Earth Engine platform [26], [27]. All cloud-free acquisitions from the 2025 dry season (June through September) were stacked and reduced to a single per-pixel median, yielding a gap-free 20-meter composite. Cloud Score and quality flags (threshold 0.60) removed cloudy and hazy pixels before the median step, so the final composite contains virtually no cloud or shadow artifacts. From each scene we retained ten spectral bands spanning visible (B2, B3, B4), red-edge (B5, B6, B7), near-infrared (B8, B8A), and shortwave infrared (B11, B12) wavelengths. All bands were resampled to a uniform 20-meter pixel grid.

Thirteen spectral indices combined from the ratio of band values to other bands yield 23 features per pixel. These indices cover five categories with different functions: vegetation (NDVI, EVI, SAVI, MSAVI, GNDVI), water (NDWI, MNDWI), buildings (NDBI, BSI), red edge (NDRE, CIRE), and moisture (NDMI, NBR). NDVI measures canopy greenness from the reflectance contrast between B8 (near infrared) and B4 (red). EVI adds a ground brightness correction and a blue band component that prevents the index from reaching saturation when leaf area is very high (a problem with NDVI over dense tropical forests). NDWI is strongly negative for vegetation but positive for water, although B3 alone shows only a slight difference between the two. Figure 3 shows the spectral characteristics of the study area. The true color RGB composite (Figure 3a) shows the visual landscape, the false color NIR composite (Figure 3b) highlights vegetation density, the NDVI (Figure 3c) reveals a vegetation gradient from water (below zero) and dense forest (above 0.8), the NDWI (Figure 3d) delimits water bodies along the Batang Hari river, and the NDBI (Figure 3e) isolates the urban core of Jambi City.

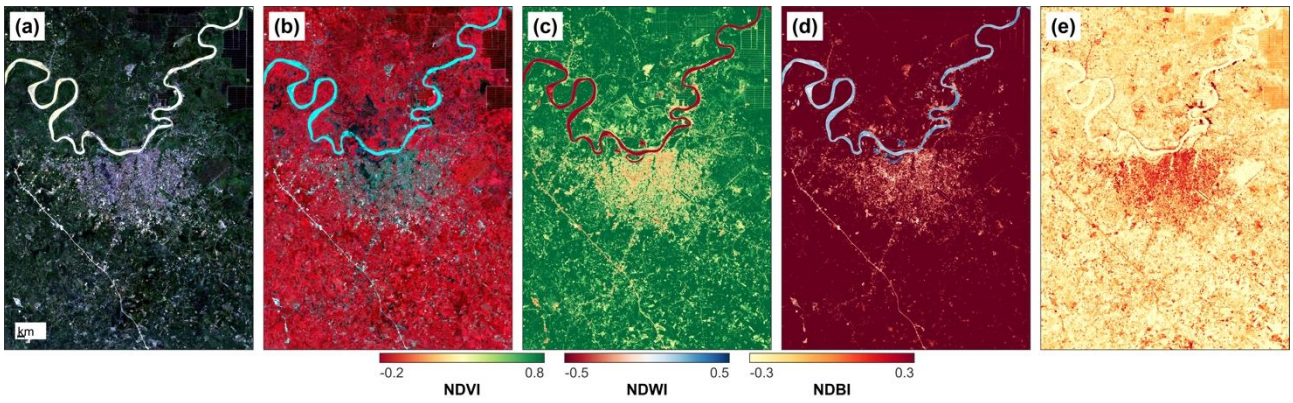

Figure 3. Spectral visualization of the study area: (a) true color RGB composite, (b) false color NIR composite with vegetation in red, (c) NDVI vegetation density, (d) NDWI water bodies, and (e) NDBI built-up areas.

Ground truth labels were extracted from Google Dynamic World using Google Earth Engine [28]. This dataset is a real-time 10-meter land cover dataset generated by a neural network trained and annotated by experts on Sentinel-2 imagery. Dynamic World was chosen because each label map is generated from the same Sentinel-2 acquisition date as the input composite. This avoids temporal mismatches that arise when reference data and satellite imagery are collected at different times. The DW assigns each pixel to nine categories: Water, Trees, Grass, Flooded Vegetation, Agricultural Crops, Shrubs/Brushes, Built-up Areas, Bare Land, and Snow/Ice. We downscaled these label resolutions from 10 meters to 20 meters using nearest neighbor interpolation so that each label cell matches a single Sentinel-2 pixel. The DW labels are model-generated and not field-verified; accuracy in this study is the degree to which they match the DW labels. If the DW incorrectly labels a particular land cover (e.g., mispredicting a rubber plantation as natural forest), then a model trained with these labels will also produce errors.

2.3. Model Architectures

Ten architectures from four model families were evaluated (Figure 4). The CNN baselines (ResNet-18 and ResNet-101 [9]) use skip connections that add the input of each block to its output, preventing the gradient from vanishing as the network grows deeper. Both were adapted for 23-channel input by repeating the

 | doi: 10.23960/pepadun.v7i3.334

pretrained 3-channel first-layer filters and rescaling, with dropout (0.3) before the classifier. Swin Transformer Tiny (Swin-T) [12] partitions the input into fixed-size local windows and computes attention independently within each window, and successive layers offset the window grid so that information flows across window borders. This windowed scheme lowers the quadratic memory cost of full self-attention while still letting pixels near window edges exchange information with their neighbors in adjacent windows.

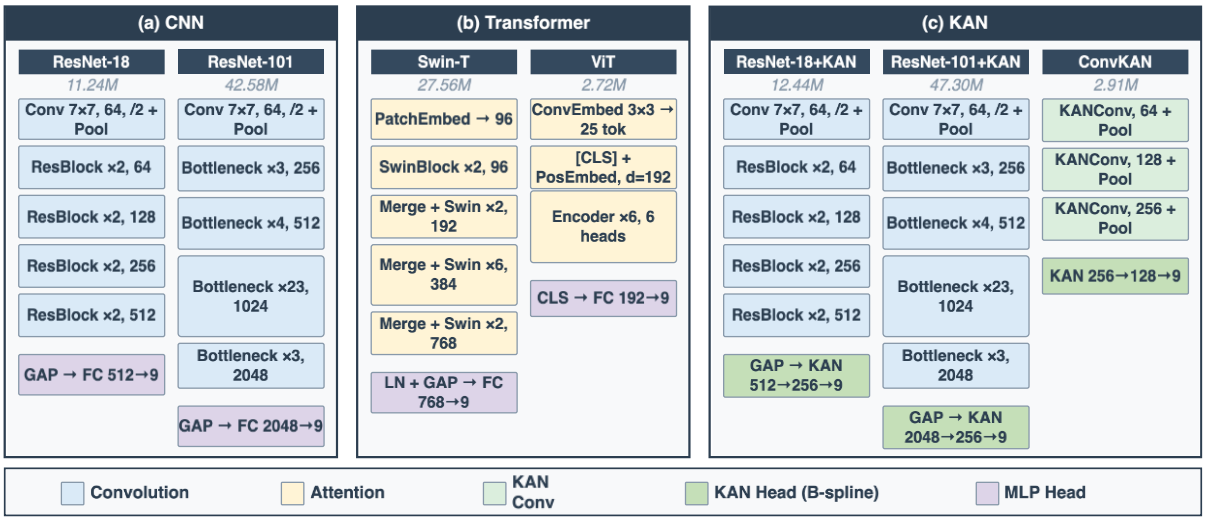

Figure 4. Architecture comparison of 7 DL models grouped by family: (a) CNN, (b) Transformer, (c) KAN. Block heights are proportional to depth. KAN models use B-spline heads (green) replacing MLP heads (purple).

A compact Vision Transformer (ViT) [10] with six encoder layers (embedding dimension 192, 6 attention heads, 2.72M parameters) was trained from scratch. It uses a convolutional embedding layer that projects each 3x3 region into a token, producing 25 tokens processed by the Transformer encoder. Two KAN integration strategies were implemented [18]. (a) *ResNet-KAN*, which replaces the fully connected classifier with two KAN layers using B-spline activations (5 control points, cubic degree); (b) *ConvKAN* [19], which replaces standard convolution filters with KAN-based spline functions in three blocks (64, 128, 256 channels) followed by a two-layer KAN classifier. Three ML baselines were also tested, all with balanced class weights. These are Random Forest [4] (200 estimators, max depth 25), Extra Trees [4] (200 estimators), and LightGBM [5] (200 boosters). For these classifiers, all 225 pixels in a 15×15 patch were averaged into a single vector of 23 values, one mean per spectral feature. This averaging removes the pixel arrangement within each patch and keeps only the mean reflectance per band.

Kolmogorov-Arnold Networks (KAN) [18] is based on the proof of the Kolmogorov-Arnold theorem. This theorem states that complex multivariate mappings can be decomposed into a finite number of sequential single-variable transformations whose outputs are combined to form the final prediction. This allows KAN to replace the fixed activation function of a traditional MLP with a learnable univariate function at each edge of the network.

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

Here, $x = (x_1, x_2, \ldots, x_n)$ represents the input features. The inner functions $\phi_{q,p}$ each process a single input variable, while the outer functions Φ_q combine their outputs. This is different from how MLPs work: MLPs apply the same fixed activation (like ReLU) to every neuron, whereas KAN gives each connection its own trainable activation curve. In our implementation, each connection uses a B-spline defined as:

$$\phi(x) = w \cdot \left(\mathrm{SiLU}(x) + \sum_{i=1}^{G+k} c_i \cdot B_{i,k}(x) \right) \tag{2}$$

 | doi: 10.23960/pepadun.v7i3.334

The scalar weight w uses Xavier initialization, while the spline coefficients c_i start near zero (drawn from $\mathcal{N}(0, 0.1^2)$) so the network initially behaves like a standard SiLU-activated layer. The basis functions $B_{i,k}(x)$ are computed using the Cox-de Boor recursion, with grid size $G = 5$ and cubic degree ($k = 3$). The grid spans the interval $[-1,1]$ with 3 extra knots on each boundary. Because each connection carries its own learnable curve instead of a single shared activation, the network can adapt different spectral bands in different ways.

For ResNet-KAN, the backbone produces a 512-dimensional feature vector (ResNet-18) or 2048-dimensional vector (ResNet-101) after global average pooling. These are fed through a KANLinear layer (512/2048 to 256 dimensions) followed by a second KANLinear layer (256 to 9 class logits). The KAN head adds approximately 1.2 million parameters for ResNet-18+KAN (total 12.44M) and approximately 4.7 million parameters for ResNet-101+KAN (total 47.30M). For ConvKAN, global average pooling produces a 256-dimensional vector that is fed through a two-layer classifier (256 to 128 to 9 class logits), producing 2.91 million total parameters, 74 percent fewer than ResNet-18.

2.4. Experimental Design and Training

We trained each model three times with different random seeds and averaged the results. The optimizer was Adam at a 1e-4 learning rate. Inverse class frequency weighting on the cross-entropy loss gave Grass and Bare Ground stronger penalties than dominant classes like Trees and Crops. None of this changed between experiments, so any difference in performance comes from the variable being tested. Table 1 defines four experiments using this fixed setup to evaluate KAN architectures. Architecture comparison ran all ten models on the same input. Feature ablation trained each model once with all 23 features and once with the 10 raw bands only. ImageNet weights were swapped for random ones because filters from three-channel RGB photographs may not transfer well to 23 multispectral input channels. Patch size went from 7×7 to 15×15. Bigger patches capture more surrounding land cover but the study area yields fewer of them.

Table 1. Experimental design and training configuration. The upper section lists the four experiment categories with the models and settings tested. The lower section lists the shared training hyperparameters.

Experiment	Models	Configurations
(1) Architecture Comparison	All 10 models	15×15 patch, 23 features
(2) Feature Ablation	ResNet-18, ResNet-18+KAN, ConvKAN, ViT	10 bands vs 23 features
(3) Transfer Learning	ResNet-18, ResNet-18+KAN, Swin-T	ImageNet vs Random init
(4) Patch Size	ResNet-18, ResNet-18+KAN, ViT	7×7, 11×11, 15×15 pixels
Optimizer / Schedule	Adam (lr=1e-4, decay=1e-4)	ReduceOnPlateau (patience=5, ×0.5)
Training	50 epochs, batch size=32	Early stopping=10, 3 seeds averaged
Class weighting	Inverse frequency	Upweight: Grass, Shrub, Bare Ground

2.5. Evaluation Metrics

F1-Weighted, F1-Macro, and OA each measure something different [29]. F1-Weighted averages per-class F1 scores after weighting each class by pixel count. This score reflects the overall map quality. F1-Macro performs the same average calculation but without weighting. Grass with a few hundred test pixels is counted the same as Trees with tens of thousands of pixels. This makes F1-Macro sensitive to rare classes that F1-Weighted and OA (Overall Accuracy) can ignore. OA is a simple calculation: the fraction of correctly labeled pixels. Trees and Crops make up over 80% of the test set. A model that gets both of these right will achieve a high OA, even if Grass and Bare Ground are completely wrong.

Cohen's Kappa coefficient [30] discounts the agreement that would occur by random chance. We rely on F1-Macro as the main ranking criterion because it exposes models that succeed only on dominant classes [31]. Individual per-class Precision, Recall, and F1-score were also reported. Precision answers "of all pixels the model labeled as class X, how many truly belong to X?", Recall answers "of all pixels that actually belong to class X, how many did the model find?", and F1 balances both measures by taking their harmonic mean.

 | doi: 10.23960/pepadun.v7i3.334

For each class i, let $TP_i = M_{ii}$ denote the true positives (correctly classified pixels), $FP_i = \sum_{j=1}^{C} M_{ji} - M_{ii}$ the false positives (pixels incorrectly predicted as class i), and $FN_i = \sum_{j=1}^{C} M_{ij} - M_{ii}$ the false negatives (pixels of class i predicted as another class), where M_{ij} is the confusion matrix element for true class i and predicted class j, and N is the total number of test pixels. The evaluation metrics are then defined as:

$$OA = \frac{\sum_{i=1}^{C} M_{ii}}{N} \tag{3}$$

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} = \frac{M_{ii}}{\sum_{j=1}^{C} M_{ji}} \tag{4}$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} = \frac{M_{ii}}{\sum_{j=1}^{C} M_{ij}} \tag{5}$$

$$F1_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{6}$$

3. RESULTS AND DISCUSSION

3.1. Architecture Comparison

All ten architectures are ranked by OA in Table 2. With 88.34% OA and 0.7043 Kappa, Swin-T leads the ranking. It correctly labels 94.7% of Trees and 83.2% of Built Area, the two classes that make up most of the test set. Swin-T processes the input through shifted-window attention. The image is split into small local windows where attention is computed separately, and the window grid then shifts so that pixels near the border of one window can exchange information with pixels in the next.

Large uniform areas like continuous forest benefit from this mechanism because neighboring pixels usually share the same class. ConvKAN ranks third in OA (86.73%) with far fewer parameters (2.91M), but it leads all models in F1-Macro (0.5870) and F1-Weighted (0.8808). This model performs best in classes with low proportions. It reaches 0.298 for Grass and 0.394 for Flooded Vegetation, and remains competitive in the dominant classes, with 0.934 for Trees and 0.891 for Built-up Area, respectively. ViT has the fewest parameters of all deep learning models (2.72 million) trained from scratch and can achieve 86.54% OA. It is highly competitive with models starting with pre-trained weights.

Table 2. Overall classification performance of all evaluated architectures. Best values per metric are shown in bold.

Model	Family	OA (%)	F1-Macro	F1-Wtd	Kappa	Params (M)	Time (s)
Swin-T	Transformer	**88,34**	0,5159	0,8757	**0,7043**	27,56	272,1
Random Forest	ML	86,75	0,4294	0,8416	0,6053	–	**0,8**
ConvKAN	KAN	86,73	**0,587**	**0,8808**	0,7023	2,91	267,3
ViT	Transformer	86,54	0,5076	0,8714	0,6722	**2,72**	115,2
Extra Trees	ML	86,28	0,4441	0,839	0,5973	–	0,3
ResNet-18+KAN	KAN	86,16	0,5555	0,8773	0,6801	12,44	162,6
ResNet-18	CNN	85,8	0,5714	0,8765	0,6821	11,24	182,8
ResNet-101	CNN	85,46	0,5588	0,8716	0,6731	42,58	934,7
LightGBM	ML	84,07	0,4691	0,8458	0,6089	–	1,9
ResNet-101+KAN	KAN	83,16	0,5099	0,8526	0,6319	47,3	610,3

OA and F1-Macro give different rankings. Swin-T tops OA at 88.34%, but its F1-Macro is only 0.5159, far below ConvKAN (0.5870) and ResNet-18 (0.5714). This gap exists because Trees and Crops make up over 80% of the test pixels. A model that labels these two classes correctly already scores a high OA, even if it misclassifies most of the rarer classes like Shrub and Crops whose reflectance overlaps with Trees.

 | doi: 10.23960/pepadun.v7i3.334

The reason ConvKAN handles minority classes better is its B-spline activations, which fit a separate non-linear curve to each input channel. Trees produce high B8 reflectance and Crops produce moderate B8, but these ranges overlap. ReLU applies the same linear slope to all values and cannot separate them. B-spline curves, on the other hand, can learn different responses for each overlapping range.

ConvKAN and ViT are also the two smallest deep learning models at 2.91M and 2.72M parameters, over 89% smaller than Swin-T (27.56M) and compact enough to run on a mid-range GPU or a CPU. Random Forest trains in under one second and reaches 86.75% OA, but its F1-Macro (0.4294) is 37% lower than ConvKAN (0.5870). Extra Trees and LightGBM behave the same way, with high OA but low F1-Macro. Without the pixel arrangement that deep learning models extract from patches, spectral features alone cannot separate minority classes in this area.

3.2. Per-Class Performance Analysis

Table 3 breaks down the F1-scores by class for all ten models across eight land cover categories (Snow/Ice was excluded due to near-zero presence in the tropical study area). All models classify Trees and Built Area reliably (F1 above 0.91 and 0.71, respectively). Snow/Ice class is excluded because the F1value is zero for all models.

Trees benefit from a large training sample and from high near-infrared reflectance (B8 above 3000 DN) that clearly separates them from non-vegetated surfaces. Rooftops and asphalt reflect strongly in the shortwave infrared (B11, B12) while vegetation absorbs those wavelengths, giving Built Area a large NDBI contrast that all models detect. Water absorbs near-infrared light (B8 near zero) while all land surfaces reflect it, producing a strongly negative NDWI that no other class shares. Minority vegetation classes (Grass, Flooded Vegetation, Crops, and Shrub/Scrub) are harder, with F1 generally below 0.50. All four classes are rare in the training set, and their reflectance values overlap with Trees in most bands. Grass and young Crops, for example, both produce NDVI between 0.3 and 0.6 and have similar red-edge values. Shrubs/Brushes are indistinctly located between Trees (NDVI above 0.7) and Crops (NDVI 0.4–0.6). No single band can clearly distinguish Shrubs from Trees or from Crops.

Table 3. Per-class F1-scores for all evaluated architectures. Best value per class is shown in bold.

Model	Water	Trees	Grass	Flood.	Crops	Shrub	Built	Bare
ResNet-18	0,734	0,931	0,229	0,31	0,32	0,432	0,886	0,727
ResNet-101	0,763	0,93	0,228	0,353	0,252	0,4	0,866	0,678
Swin-T	**0,799**	**0,947**	0,086	0,381	0,019	0,448	0,832	0,615
ViT	0,652	0,94	0,17	0,185	0,116	0,384	0,873	**0,741**
ResNet-18+KAN	0,792	0,935	0,208	0,37	0,241	0,411	0,88	0,606
ResNet-101+KAN	0,746	0,917	0,238	0,262	0,129	0,373	0,835	0,58
ConvKAN	0,736	0,934	**0,298**	**0,394**	**0,296**	**0,481**	**0,891**	0,667
Random Forest	0,767	0,932	0,032	0,154	0	0,209	0,732	0,609
Extra Trees	0,77	0,931	0,029	0,231	0,019	0,191	0,716	0,667
LightGBM	0,731	0,927	0,137	0,177	0,073	0,274	0,767	0,667

Looking at Table 3 for each class, the easiest classes to predict were Water, Trees, and Built-up Area. Each model handled them well. Swin-T scored the highest on Water and Trees at 0.799 and 0.947, respectively. These two classes cover large, uniform areas where windowed attention works well. The hardest classes to predict were Grass, Flooded Vegetation, Agricultural Crops, and Shrubs. Their NDVI values were all between 0.3 and 0.6, and at 20-meter pixels most models considered them to be the same. ConvKAN separated them better than the other models. It scored 0.298 on Grass, more than three times Swin-T's score of 0.086. ConvKAN also excelled on Flooded Vegetation, Shrubs, and Built-up Area. ViT was the only model that handled Bare Land well, with a score of 0.741. Bare ground has a characteristic SWIR-bright and NIR-dark spectrum that is captured by the self-attention of the surrounding vegetation.

© 2026 The Authors | doi: 10.23960/pepadun.v7i3.334

Each ML classifier received a single vector of mean band values per patch, not the original image (Section 2.3). That averaging erases all spatial patterns. A rice paddy with neat planted rows and a forest patch with scattered crowns end up with nearly identical mean reflectance, separated by under five percent across most bands. The scores in Table 3 confirm this. Random Forest completely misses Crops and scores near zero on Grass. Extra Trees and LightGBM do about the same. KAN models train on the full patch and keep spatial patterns intact. On top of that, their B-spline activations learn a separate nonlinear mapping for each spectral band. One mapping responds to the red-edge slope of Grass. Other stays flat for young Crops that fall in a nearby reflectance range. ReLU applies a single fixed slope to every value and cannot adapt per band. This is the main reason KAN produces more balanced vegetation scores than the ML baselines.

3.3. Feature Ablation Study

Table 4 quantifies how much the 13 derived spectral indices contribute to each architecture's accuracy. Each model was trained twice, once with all 23 features and once with only the 10 raw bands. Comparing the two runs shows how much the pre-computed indices help or hurt each architecture. The question is whether pre-computing band ratios such as NDVI and NDWI as additional input channels helps each architecture, or whether the models can learn equivalent features from the raw bands alone.

KAN-based architectures benefit more from spectral indices than standard CNNs or Transformers. ConvKAN shows the largest gain, with F1-Macro rising by 0.0713 (+13.8%) and OA by 1.80 percentage points. ResNet-18+KAN follows a similar trend (+2.32% OA), while standard ResNet-18 shows a marginal OA decrease (minus 0.36%) when indices are added. B-spline activations in KAN can combine pre-computed indices (e.g., multiply NDVI by NDMI) into compound features that reveal differences between irrigated and rain-fed vegetation. Standard convolutions with fixed ReLU treat these extra channels the same way as raw bands and gain less from them.

Table 4. Feature ablation results comparing spectral bands only (10 features) versus bands combined with spectral indices (23 features). Difference is computed as 23-feature result minus 10-feature result.

Model	Features	OA (%)	F1-Macro	F1-Wtd	Kappa
ConvKAN	Bands only (10)	85,53	0,5178	0,8661	0,6623
ConvKAN	Bands + Indices (23)	87,33	0,5891	0,8863	0,7102
ConvKAN	**Δ**	**+1.80**	**+0.0713**	**+0.0202**	**+0.0479**
ResNet-18+KAN	Bands only (10)	84,91	0,5432	0,8685	0,6639
ResNet-18+KAN	Bands + Indices (23)	87,23	0,5625	0,8809	0,6948
ResNet-18+KAN	**Δ**	**+2.32**	**+0.0193**	**+0.0124**	**+0.0309**
ResNet-18	Bands only (10)	85,44	0,5471	0,873	0,6652
ResNet-18	Bands + Indices (23)	85,08	0,5759	0,8687	0,6672
ResNet-18	**Δ**	**−0.36**	**+0.0288**	**−0.0043**	**+0.0020**
ViT	Bands only (10)	86,85	0,5331	0,8726	0,6878
ViT	Bands + Indices (23)	83,66	0,4822	0,8454	0,6263
ViT	**Δ**	**−3.19**	**−0.0509**	**−0.0272**	**−0.0616**

ViT is the exception, as adding spectral indices *decreases* OA by 3.19 percentage points and F1-Macro by 0.0509. Multi-head self-attention weighs every token against every other token, so ViT can internally learn to divide B8 by B4 (the NDVI formula) without needing it as an explicit input. When the same ratio is also supplied as a pre-computed channel, the redundancy adds noise to the optimization landscape. B-spline activations work differently and can combine two indices into a joint discriminant. For instance, a spline can learn that pixels with both high NDVI and high NDMI are irrigated crops, while pixels with similar NDVI but low NDMI are rain-fed grassland. The Kappa deltas show the same trend. ConvKAN gains +0.0479 when indices are added, ResNet-18 gains only +0.0020, and ViT drops by 0.0616. KAN-based architectures should therefore be trained with spectral indices included, while ViT-based models perform better with only the 10 raw bands.

 | doi: 10.23960/pepadun.v7i3.334

3.4. Transfer Learning Analysis

Table 5 compares ImageNet-pretrained initialization against random initialization. Models pretrained on millions of natural RGB photographs often perform better on downstream tasks, but 20 of the 23 Sentinel-2 input channels (red-edge, NIR, SWIR) lie outside the visible spectrum and have no counterpart in the original ImageNet data. The question is whether this spectral mismatch cancels the benefit of pretraining. Swin-T benefits from ImageNet pretraining (+1.24% OA, +0.0384 F1-Macro) [14]. Its attention layers detect edges between adjacent land cover patches and repeating texture patterns (e.g., row spacing in plantations), and these geometric cues remain valid regardless of how many spectral channels the image has. Both ResNet-18 and ResNet-18+KAN perform better with random initialization. The effect is strongest for ResNet-18+KAN (+2.49% OA, +0.1321 F1-Macro), because the first convolutional layer in a CNN is tightly coupled to the input channels. Pretrained filters tuned for three visible bands become a poor initialization when repeated across 23 multispectral channels, especially for the 20 non-RGB channels. B-spline activations in KAN layers amplify the mismatch, as each spline must reshape itself jointly with the backbone filters during training. Starting from pretrained filters that encode RGB-specific patterns forces the splines into a configuration far from the optimum. For multispectral imagery beyond the visible spectrum, KAN-based models should therefore be trained from scratch rather than initialized with ImageNet weights.

Table 5. Transfer learning results comparing ImageNet pretrained initialization versus random initialization. Bold values indicate the better initialization strategy for each model.

Model	Init.	OA (%)	F1-Macro	F1-Wtd	Kappa	Time (s)
Swin-T	ImageNet	**87,04**	**0,5502**	**0,8797**	**0,6899**	470,9
Swin-T	Random	85,8	0,5118	0,8622	0,6626	338,7
ResNet-18	ImageNet	84,36	0,5466	0,8659	0,6551	236,7
ResNet-18	Random	**84,93**	**0,5691**	**0,8714**	**0,6683**	254,2
ResNet-18+KAN	ImageNet	82,92	0,4394	0,8431	0,6063	239,8
ResNet-18+KAN	Random	**85,41**	**0,5715**	**0,8753**	**0,6705**	228,2

3.5. Patch Size Sensitivity

The effect of input patch size on classification performance is shown in Table 6. Patch size controls two factors simultaneously, namely the spatial extent visible to the model and the number of training samples that can be extracted from the study area. A 7×7 patch covers 140 by 140 meters and contains only 49 pixels, so the model sees very little of the surrounding area. From the study area we extracted 22,500 patches at 7×7 and only 4,175 at 15×15. The difference matters. A 7×7 patch is 140 by 140 meters, just 49 pixels. The model has almost no view of the area around the center pixel. A 15×15 patch is 300 by 300 meters and lets the model see nearby rivers, plantations, and forest edges.

Patch size involves a trade-off between the number of training samples and how much surrounding context each sample contains. Smaller patches produce more training samples. At 7×7 the training set contains 22,500 patches, but at 15×15 only 4,175 can be extracted from the same study area. Larger patches let the model see the surrounding land cover, such as plantation grids, river banks, and forest edges. Not all models respond to patch size the same way. ResNet-18+KAN at 7×7 reaches 87.98% OA and 0.7120 Kappa, the highest in this study. The five times larger training set at that patch size drives the result. ResNet-18 is different. Its F1-Macro climbs from 0.4525 at 7×7 to 0.5228 at 11×11 to 0.5514 at 15×15. For ResNet-18, seeing the surrounding land cover at 15×15 helps separate minority classes more than having five times as many training patches at 7×7. ResNet-18+KAN and ViT both reach their best F1-Macro at 11×11, scoring 0.5531 and 0.5583. A 220-meter window is enough for B-spline and self-attention models to distinguish Shrub from Trees.

Table 6. Patch size sensitivity analysis for ResNet-18, ResNet-18+KAN, and ViT. Patch sizes correspond to 140m, 220m, and 300m ground coverage at 20m resolution.

Model	Patch	OA (%)	F1-Macro	F1-Wtd	Kappa	Samples	Time (s)
ResNet-18	7×7	**86,14**	0,4525	0,864	**0,6711**	22500	290,9

 | doi: 10.23960/pepadun.v7i3.334

Model	Patch	OA (%)	F1-Macro	F1-Wtd	Kappa	Samples	Time (s)
ResNet-18	11×11	85,98	0,5228	**0,8738**	0,6695	9470	430,3
ResNet-18	15×15	84,72	**0,5514**	0,867	0,6602	4175	251
ResNet-18+KAN	7×7	**87,98**	0,5069	**0,8834**	**0,712**	22500	1048,7
ResNet-18+KAN	11×11	86,57	**0,5531**	0,8791	0,6926	9470	570,2
ResNet-18+KAN	15×15	81,94	0,4217	0,8271	0,5657	4175	129,7
ViT	7×7	**87,37**	0,5417	**0,886**	**0,7125**	22500	978,2
ViT	11×11	85,02	**0,5583**	0,8684	0,6714	9470	311,2
ViT	15×15	86,87	0,5276	0,8758	0,6864	4175	116,7

No single patch size works best for every model and metric. Smaller 7×7 patches tend to give the highest OA thanks to more training samples (22,500 versus 4,175), while 11×11 or 15×15 patches give better F1-Macro because they capture enough surrounding context. When per-class balance matters most, 11×11 patches (220 by 220 meters) are a practical middle ground for B-spline and self-attention architectures

3.6. Spatial Prediction Comparison

Figure 5 shows the prediction maps from all seven deep learning models next to the Dynamic World ground truth. OA and F1 do not measure how noisy a map looks or how sharp the boundaries between classes are, so visual comparison adds information that the numbers in Tables 2 and 3 cannot provide.

The Ground Truth panel (Figure 5a) is mostly dark green Trees covering the western and southern parts of the map. The Batang Hari river winds in blue through the upper portion from west to east. Jambi City appears as a large magenta Built Area cluster in the center-right. Olive Crops and scattered pink Shrub/Scrub patches sit between the city and the forest. Narrow cyan strips of Flooded Vegetation line the river banks. The map reads roughly as forest on the left, farmland in the middle, and city on the right.

In the forest regions, both ResNet variants scatter wrong labels across the canopy. ResNet-18 (Figure 5b, OA = 77.64%) reproduces the broad layout but labels isolated forest pixels as Crops or Shrub. ResNet-101 (Figure 5c, OA = 74.55%) is worse, with more frequent class switching between adjacent pixels. The urban edge of Jambi City is fragmented in both maps, and the Batang Hari river breaks into disconnected segments where the channel narrows. ResNet-101 also generates a large Crops/Bare Ground artifact near the center of the map that does not appear in the ground truth. Going from 18 layers to 101 brings no improvement for this study area.

Panel 5d (Swin-T, OA = 77.22%) stands out from the ResNet panels right away. The forest noise is gone and the Batang Hari river runs unbroken from west to east. The boundary around Jambi City has fewer misclassified pixels at its edge. There are also fewer misclassified pixels at the edges between land cover types. Panel 5e (ViT, OA = 85.75%) is the closest match to the ground truth. Forest coverage is solid and the urban perimeter matches the ground truth outline. Bare Ground patches in the south show up correctly in brown. These patches do not appear in any other model's output. Most models call them Shrub. ViT's F1 on Bare Ground is 0.741, the highest in Table 3.

The two ResNet-KAN models (Figures 5f and 5g) do not improve much over their plain ResNet counterparts. ResNet-18+KAN (OA = 78.48%) has slightly less forest noise than ResNet-18 but still falls short of the Transformer results. ResNet-101+KAN (OA = 76.21%) is the most fragmented panel of all eight, with misclassified pixels scattered across forest and agricultural areas alike. Adding a KAN head to a deeper backbone does not reduce the noise. ConvKAN (Figure 5h, OA = 78.45%) is different. Its map is as smooth as the Swin-T result. The urban boundary is clean and the forest is free of scattered wrong labels. Transitions between Trees and Crops are better defined than in any ResNet variant.

The river turns out to be a good indicator of overall map quality. Swin-T, ViT, and ConvKAN all render it as a continuous blue line. These three models also produce cleaner class boundaries in the rest of the map. ResNet-18, ResNet-101, and ResNet-101+KAN break the river into segments. These models show more noise

 | doi: 10.23960/pepadun.v7i3.334

everywhere else as well. ConvKAN and ViT achieve this visual quality with only 2.91M and 2.72M parameters, roughly ten times fewer than Swin-T at 27.56M. The difference between ConvKAN and ResNet-101+KAN is that spline-based convolution filters operate at every layer, not just at the final classification stage. A model with 2–3 million parameters can therefore produce maps as clean as a 27-million parameter model, as long as spline activations are used in every layer instead of only at the classifier head.

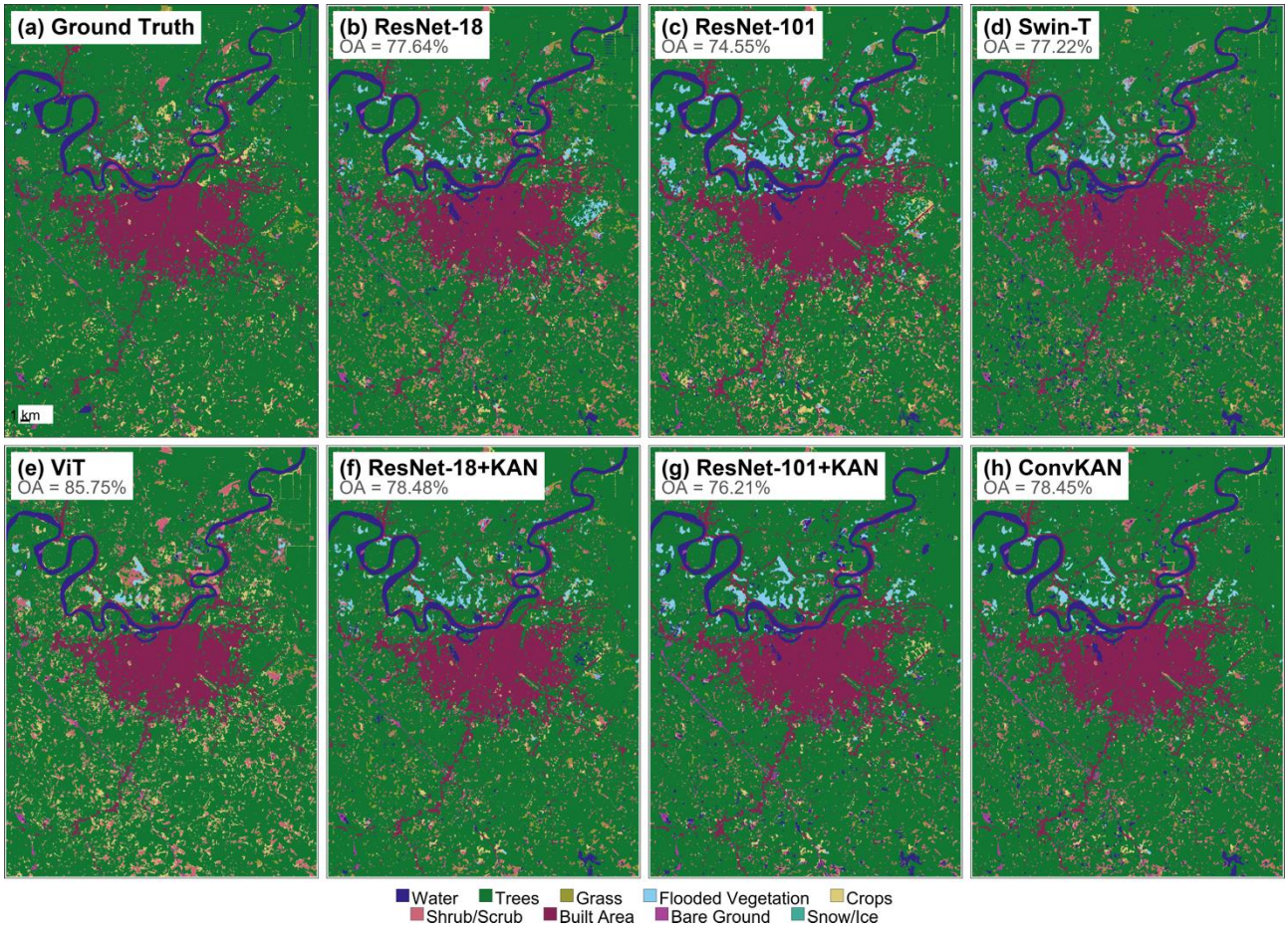

Figure 5. Spatial prediction maps for (a) Ground Truth, (b) ResNet-18, (c) ResNet-101, (d) Swin-T, (e) ViT, (f) ResNet-18+KAN, (g) ResNet-101+KAN, and (h) ConvKAN over the Jambi City area.

To contextualize our results, we compare our models against four published Sentinel-2 land cover classification studies (Table 7). Our ConvKAN achieves competitive accuracy with a significantly smaller parameter budget compared to established global products. While higher overall accuracies are reported on balanced benchmarks like EuroSAT (Cheon 2024), those results are not directly comparable to our imbalanced, spectrally overlapping study area. In this context, ConvKAN's leading F1-Macro score demonstrates its effectiveness in classifying minority land cover types that standard benchmarks often overlook.

Table 7. Comparison with published land cover classification studies using Sentinel-2 imagery.

Study	Dataset	Region	Method	OA (%) (Classes)	Parameters
Our work (ConvKAN)	Sentinel-2 (Jambi) Own test: 22,500 patches	Indonesia (tropical)	ConvKAN (B-spline) F1-Macro: 0.5870 Also: Swin-T, ResNet-18+KAN	86,73 (9)	2.91M (89.4% fewer than Swin-T)
Brown et al. [28] (2022)	Sentinel-2 (global) Dynamic World dataset	Global	FCNN (DeepLab-like) Expert consensus validation Nature Scientific Data 9:251	73,8 (9)	~0.3M (~100x smaller than U-Net)

 | doi: 10.23960/pepadun.v7i3.334

Study	Dataset	Region	Method	OA (%) (Classes)	Parameters
Karra et al. [27] (2021)	Sentinel-2 (global) Esri 10m LULC product	Global	U-Net / CNN Stratified random validation IEEE IGARSS 2021	86 (10)	~31millions (U-Net backbone)
Fawzy & Barsi [32] (2025)	VHR multispectral satellite Urban land cover	Hungary (urban)	KAN vs SNN KAN: 10-neuron mid-layer KAN: 88.89% OA, SNN: 87.84%	88,89 (7)	Reduced (10-neuron mid-layer)
Cheon [20] (2024)	EuroSAT (Sentinel-2) 27,000 balanced images	Global (benchm)	ConvNeXt + KAN 96% OA by epoch 2 Nodes reducible 256 to 32 arXiv:2406.00600	96 (10)	Fewer params than MLP

4. CONCLUSIONS

We compared KAN-based architectures against CNN, Transformer, and machine learning baselines for Sentinel-2 land cover mapping in Jambi, Indonesia. The comparison covered ten architectures from four families, all evaluated on the same test set with nine Dynamic World land cover classes and four controlled ablation experiments. Three findings stand out. First, ConvKAN achieves the highest F1-Macro (0.5870) among all ten architectures with 89\% fewer parameters than Swin Transformer (2.91M vs 27.56M), because learnable B-spline curves can draw non-linear decision boundaries between classes like oil palm (NDVI 0.7) and secondary forest (NDVI 0.8) that fixed ReLU slopes cannot separate. Second, KAN-based architectures benefit more from spectral indices than CNNs or Transformers, whereas adding indices to Vision Transformer decreases its accuracy, because self-attention already learns to divide and subtract bands (e.g., computing NDVI from B8 and B4) without external index channels. Third, ResNet-KAN performs better with random initialization than with ImageNet pretraining, because pretrained filters tuned for three RGB channels are a poor starting point when 20 of the 23 input channels (red-edge, NIR, SWIR) have no RGB equivalent. Several limitations apply. Dynamic World labels are model-derived rather than field-verified, minority vegetation categories are under-represented in the training set, and only one tropical study area in Sumatra was tested. Future work should apply KAN to hyperspectral imagery to evaluate whether B-spline activations benefit from the hundreds of narrow, contiguous wavelength channels that hyperspectral sensors provide. A hybrid design combining windowed self-attention for spatial patterns (e.g., plantation row spacing) with B-spline edge activations for overlapping reflectance ranges could outperform either approach alone. Field-surveyed reference data would provide more reliable accuracy estimates and test whether these findings generalize to other tropical in all over Indonesia regions.

LITERATURE

[1] S. Zhao, K. Tu, S. Ye, H. Tang, Y. Hu, and C. Xie, “Land Use and Land Cover Classification Meets Deep Learning: A Review,” *Sensors*, vol. 23, no. 21, p. 8966, 2023, doi: 10.3390/s23218966.

[2] D. Phiri, M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama, and M. Ranagalage, “Sentinel-2 Data for Land Cover/Use Mapping: A Review,” *Remote Sens. (Basel).*, vol. 12, no. 14, p. 2291, 2020, doi: 10.3390/rs12142291.

[3] D. Montero, C. Aybar, M. D. Mahecha, F. Martinuzzi, M. Söchting, and S. Wieneke, “A standardized catalogue of spectral indices to advance the use of remote sensing in Earth system research,” *Sci. Data*, vol. 10, p. 197, 2023, doi: 10.1038/s41597-023-02096-0.

[4] S. Talukdar, P. Singha, S. Mahato, Shahfahad, S. Pal, Y. A. Liou, and A. Rahman, “Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations—A Review,” *Remote Sens. (Basel).*, vol. 12, no. 7, p. 1135, 2020, doi: 10.3390/rs12071135.

 | doi: 10.23960/pepadun.v7i3.334

[5] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3149–3157.

[6] A. Vali, S. Comai, and M. Matteucci, "Deep Learning for Land Use and Land Cover Classification Based on Hyperspectral and Multispectral Earth Observation Data: A Review," *Remote Sens. (Basel).*, vol. 12, no. 15, p. 2495, 2020, doi: 10.3390/rs12152495.

[7] T. Boston, A. Van Dijk, and R. Thackway, "Convolutional Neural Network Shows Greater Spatial and Temporal Stability in Multi-Annual Land Cover Mapping Than Pixel-Based Methods," *Remote Sens. (Basel).*, vol. 15, no. 8, p. 2132, 2023, doi: 10.3390/rs15082132.

[8] Q. Yuan, H. Shen, T. Li, Z. Li, S. Li, Y. Jiang, H. Xu, W. Tan, Q. Yang, J. Wang, J. Gao, and L. Zhang, "Deep learning in environmental remote sensing: Achievements and challenges," *Remote Sens. Environ.*, vol. 241, p. 111716, 2020, doi: 10.1016/j.rse.2020.111716.

[9] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, 2021, doi: 10.1016/j.isprsjprs.2020.12.010.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2010.11929*, 2021.

[11] Y. Bazi, L. Bashmal, M. M. Al Rahhal, R. Al Dayil, and N. Al Ajlan, "Vision Transformers for Remote Sensing Image Classification," *Remote Sens. (Basel).*, vol. 13, no. 3, p. 516, 2021, doi: 10.3390/rs13030516.

[12] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10012–10022. doi: 10.1109/ICCV48922.2021.00986.

[13] A. A. Aleissaee, A. Kumar, R. M. Anwer, S. Khan, H. Cholakkal, G. S. Xia, and F. S. Khan, "Transformers in Remote Sensing: A Survey," *Remote Sens. (Basel).*, vol. 15, no. 7, p. 1860, 2023, doi: 10.3390/rs15071860.

[14] R. Naushad, T. Kaur, and E. Ghaderpour, "Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study," *Sensors*, vol. 21, no. 23, p. 8083, 2021, doi: 10.3390/s21238083.

[15] R. Sugumar and D. Suganya, "Satellite imagery for land cover classification using machine learning techniques," *Multimed. Tools Appl.*, pp. 1–26, 2025, doi: 10.1007/S11042-025-20928-6.

[16] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, 2019, doi: 10.1016/j.isprsjprs.2019.04.015.

[17] X. X. Zhu, D. Tuia, L. Mou, G. S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, 2017, doi: 10.1109/MGRS.2017.2762307.

[18] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-Arnold Networks," *arXiv preprint arXiv:2404.19756*, 2024, doi: 10.48550/arXiv.2404.19756.

[19] A. D. Bodner, A. S. Tepsich, J. N. Spolski, and S. Pourteau, "Convolutional Kolmogorov-Arnold Networks," *arXiv preprint arXiv:2406.13155*, 2024.

[20] M. Cheon, "Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing," *arXiv preprint arXiv:2406.00600*, 2024, doi: 10.48550/arXiv.2406.00600.

 | doi: 10.23960/pepadun.v7i3.334

[21] N. Firsov, V. Lobanov, E. Myasnikov, R. Khabibullin, and A. Nikonorov, "HyperKAN: Kolmogorov-Arnold Networks Make Hyperspectral Image Classifiers Smarter," *Sensors*, vol. 24, no. 23, p. 7683, 2024, doi: 10.3390/s24237683.

[22] A. Jamali, S. K. Roy, D. Hong, B. Lu, and P. Ghamisi, "How to Learn More? Exploring Kolmogorov-Arnold Networks for Hyperspectral Image Classification," *Remote Sens. (Basel).*, vol. 16, no. 21, p. 4015, 2024, doi: 10.3390/rs16214015.

[23] Y. Wang, X. Yu, Y. Gao, J. Sha, J. Wang, S. Yan, K. Qin, Y. Zhang, and L. Gao, "SpectralKAN: Weighted Activation Distribution Kolmogorov–Arnold Network for Hyperspectral Image Change Detection," *Pattern Recognit.*, vol. 175, p. 113042, 2026, doi: https://doi.org/10.1016/j.patcog.2026.113042.

[24] E. Rustiadi, A. E. Pravitasari, R. A. Priatama, J. Siregar, J. Junaidi, Z. Zulgani, and R. I. Sholihah, "Regional Development, Rural Transformation, and Land Use/Cover Changes in a Fast-Growing Oil Palm Region: The Case of Jambi Province, Indonesia," *Land (Basel).*, vol. 12, no. 5, p. 1059, 2023, doi: 10.3390/land12051059.

[25] I. L. Sari, C. J. Weston, G. J. Newnham, and L. Volkova, "Developing Multi-Source Indices to Discriminate between Native Tropical Forests, Oil Palm and Rubber Plantations in Indonesia," *Remote Sens. (Basel).*, vol. 14, no. 1, p. 3, 2022, doi: 10.3390/rs14010003.

[26] H. Tamiminia, B. Salehi, M. Mahdianpari, L. Quackenbush, S. Adeli, and B. Brisco, "Google Earth Engine for geo-big data applications: A meta-analysis and systematic review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 164, pp. 152–170, 2020, doi: 10.1016/j.isprsjprs.2020.04.001.

[27] K. Karra, C. Kontgis, Z. Statman-Weil, J. C. Mazzariello, M. Mathis, and S. P. Brumby, "Global land use/land cover with Sentinel-2 and deep learning," *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 4704–4707, 2021, doi: 10.1109/IGARSS47720.2021.9553499.

[28] C. F. Brown, S. P. Brumby, B. G. Williams, T. Birch, S. B. Hyde, J. Mazzariello, W. Czerwinski, V. J. Pasquarella, R. Haertel, S. Ilyushchenko, K. Schwehr, M. Weisse, F. Stolle, C. Hanson, O. Guinan, R. Moore, and A. M. Tait, "Dynamic World, Near real-time global 10 m land use land cover mapping," *Sci. Data*, vol. 9, p. 251, 2022, doi: 10.1038/s41597-022-01307-4.

[29] A. E. Maxwell, T. A. Warner, and L. A. Guillén, "Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review," *Remote Sens. (Basel).*, vol. 13, no. 13, p. 2450, 2021, doi: 10.3390/rs13132450.

[30] G. M. Foody, "Explaining the unsuitability of the kappa coefficient in the assessment and comparison of the accuracy of thematic maps obtained by image classification," *Remote Sens. Environ.*, vol. 239, p. 111630, 2020, doi: 10.1016/j.rse.2019.111630.

[31] S. Farhadpour, T. A. Warner, and A. E. Maxwell, "Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices," *Remote Sens. (Basel).*, vol. 16, no. 3, p. 533, 2024, doi: 10.3390/rs16030533.

[32] M. Fawzy and Á. Barsi, "VHR Multispectral Satellite Image Classification with Kolmogorov-Arnold Networks for Urban Applications," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-G-2025, pp. 245–252, 2025, doi: 10.5194/isprs-annals-X-G-2025-245-2025.