remote sensing

MDPI

Article

Comparison of Lightweight Deep Neural Networks for Landsat Time-Series Land Use and Land Cover Classification over the Conterminous United States

Zhixin Wang, Giorgos Mountrakis * and Ahmadreza Safaeinia

Department of Environmental Resources Engineering, State University of New York College of Environmental Science and Forestry, Syracuse, NY 13210, USA; zwang127@esf.edu (Z.W.)
* Correspondence: gmountrakis@esf.edu

Highlights

What are the main findings?

- Simple Recurrent Unit-based lightweight models consistently outperformed traditional classifiers with small model sizes.
- From the tested SRU models, MobileNet offered the greatest improvement.

What are the implications of the main findings?

- When model complexity is constrained due to limited labels or computing resources, selecting lightweight models can improve the trade-off between model efficiency and classification accuracy.

Abstract

Accurate and timely land cover and land use (LCLU) classification from medium-spatial-resolution optical time-series data is essential for large-scale environmental monitoring. lightweight deep neural networks (DNNs) offer reduced computational and memory requirements, enabling efficient deployment in resource-constrained scenarios. While popular in computer vision tasks, their ability to simultaneously model spatial, spectral, and temporal information for medium-resolution optical time series is understudied. This study addresses this gap by evaluating seven existing lightweight models spanning four architectural families: convolutional and recurrent hybrids, convolutional and transformer hybrids, 3D convolutional models, and video transformers against a traditional hybrid convolutional transformer (CNNTransformer) benchmark across the Conterminous United States (CONUS). Models are trained on 500,000 Landsat time-series samples with 25 repetitions and evaluated across five model sizes (3k, 5k, 10k, 25k, and 50k parameters) to assess both accuracy and stability. Results show that Simple Recurrent Unit (SRU)-based lightweight hybrids provide the best performance. Specifically, MobileNetSRU consistently outperformed the benchmark at small-to-moderate model sizes (3k–15k), achieving peak relative improvement gains of ~2.5–7.5% at 7.5k parameters. MobileNetSRU also demonstrated superior robustness in limited-data scenarios (50k training samples), particularly for spectrally stable classes like water and bare land. This study reveals that the inherent inductive bias of recurrent-based lightweight models aligns more effectively with the sequential phenology of satellite data than more flexible, data-hungry attention mechanisms at small parameter scales. These findings suggest that strategically matching architectural priorities to temporal data structures can significantly reduce the trade-off between model efficiency and classification accuracy in scalable Earth-observation workflows.

Academic Editors: Liming Zhou, Hao Zhu and Biao Hou

Received: 2 March 2026
Revised: 12 May 2026
Accepted: 21 May 2026
Published: 1 June 2026

Keywords: lightweight deep neural networks; convolutional; recurrent; transformer; land cover land use; time-series; medium resolution; conterminous U.S.

1. Introduction

Accurate and timely land cover land use (LCLU) classification is fundamental to monitoring environmental changes [1–3], managing natural resources [4,5], and informing sustainable policy [6–8]. Although land cover (LC) refers to the physical materials on the Earth's surface and land use (LU) describes how humans utilize those surfaces, LC inherently provides the observable foundation from which many LU categories are inferred. In remote sensing (RS), the combined term LCLU is therefore commonly adopted to reflect both the physical surface characteristics and their functional interpretation. Following this convention, the term LCLU were used throughout this study, while recognizing that the classification task is primarily driven by LC information derived from Landsat spectral–temporal observations.

The Landsat archive, with its continuous, global-scale data collected since 1972, represents an unparalleled resource for large-area and long-term analysis of the Earth's surface. The Landsat data's rich spatial (30 m), spectral (multiple bands), and temporal (biweekly) dimensions provide a robust foundation for tracking complex land surface dynamics.

Harnessing the full potential of this extensive spatial and temporal archive for LCLU classification presents a significant computational challenge. Processing large-area, long-period time-series data requires models that can efficiently integrate spatial, spectral, and temporal information simultaneously. Traditional deep learning (DL) models, particularly transformers, have become popular for LCLU classification; however, they are typically computationally intensive [9,10]. The high parameter counts and operational costs of these "heavy" models can create a bottleneck for large-scale deployment, operational monitoring, and processing on resource-constrained devices, such as mobile phones and edge devices [11,12].

Lightweight deep neural networks (DNNs) are efficient DL models, architecturally optimized to reduce computational load, memory usage, and parameter count, making them suitable for deployment on edge devices with limited resources [11,13]. Lightweight models enable on-device processing, ensuring data privacy and ownership by eliminating the need to transmit sensitive information to the cloud. By keeping processing local, these architectures reduce communication overhead and network dependency, making them ideal for secure, real-time decentralized monitoring. Lightweight DNN models have been applied in LCLU classification by offering a good balance between efficiency and accuracy. These lightweight models primarily focus on two tasks: scene classification and hyperspectral classification. Recent studies have demonstrated the effectiveness of lightweight convolutional neural networks (CNNs) in achieving accuracy comparable to state-of-the-art heavy models in scene classification. Scene classification refers to assigning a single label to an entire scene or image. These models are typically evaluated on standard public datasets, including AID [14], NWPU [15], and UC Merced [16], which consist of satellite and aerial RGB images with resolutions ranging from 0.2 m to 30 m. For instance, Tong et al. [17] developed a lightweight DenseNet model based on DenseNet121, which performed comparably to heavy CNNs across these datasets. Liu and Bai [18] further improved classification by incorporating a statistically independent Gaussian noise-based feature augmentation (IGNA) module into a lightweight CNN, outperforming MobileNetV2 on the same datasets. Additionally, knowledge distillation has been used to enhance lightweight models. Song et al. [19] employed an ensemble of EfficientNet-b0

and EfficientNet-b3 in a teacher–student knowledge-distillation framework, achieving superior performance on the AID and NWPU datasets compared to other lightweight and heavy CNNs.

Hyperspectral classification generally involves semantic segmentation, i.e., assigning a label to each pixel in an image. Lightweight models, both CNN-based and transformer-based, have been proposed to extract spatial and spectral features from hyperspectral datasets like Indian Pines [20], Houston 2013 [21], Pavia University [22], and Salinas [23]. For example, researchers have successfully utilized hierarchical residual strategies [24], 1D convolutional layers within transformers [25], and hybrid CNN–transformer designs [26] across these datasets. Furthermore, gradient-based Network Architecture Search (NAS) has been employed to automatically design efficient attention networks that outperform traditional state-of-the-art models [27].

While lightweight models have advanced for scene and hyperspectral classification, their application to high-spatial-resolution optical imagery in LCLU classification remains relatively underexplored. Recent studies have successfully adapted architectures like MobileNetV2 and lightweight UNet for specific tasks, including water body and cropland classification, often outperforming heavier 2D CNNs and Vision Transformers [28,29]. Similarly, Synthetic Aperture Radar (SAR) imagery was primarily used by lightweight models for object detection [30,31]. Lightweight models like MobileNet and SegNet have shown promise for specialized LCLU tasks such as green-tide detection [32].

Medium-spatial-resolution optical imagery, the focus of this work, has been sparsely explored by lightweight models in LCLU classification. Among the medium-spatial-resolution optical sensors, Sentinel-2 (10 m) and Landsat (30 m) are mostly used for their free availability and continuous global coverage, compared to other sensors such as ASTER (15 m). Here, lightweight models utilizing Sentinel-2 were first presented, followed by those employing Landsat data. Mazzia et al. [33] combined a Recurrent Neural Network (RNN) with CNN using one-year Sentinel-2 temporal data for crop classification in a north–central part of Italy. Results showed the lightweight model (96%) with ~31k parameters was better than traditional machine learning methods such as Support Vector Machine (SVM) (80%), and Random Forest (RF) (78%). Garnot and Landrieu [34] designed a lightweight Temporal Self-Attention model with ~150k parameters for crop classification using one-year Sentinel-2 time series and the lightweight model (94.3%) achieved comparable accuracy to the traditional Temporal Attention Encoder (TAE) (94.2%) and outperformed TempCNN (93.3%). Corbane et al. [35] developed a simple CNN with ~1.4M parameters for built-up classification using Sentinel-2 images, although the model was not compared to others. Arrechea-Castillo et al. [36] used a simple CNN based on LeNet using two Sentinel-2 images for LCLU classification in a sub-basin in Colombia. The model achieved the highest accuracy (96.5%) compared to six traditional DL architectures (e.g., AlexNet (96.0%) and ResNet (96.3%)) and it also outperformed an efficient model, EfficientNet (94.9%). Papoutsis et al. [37] designed a lightweight CNN with ~30k parameters based on EfficientNet with Sentinel-2 images for multi-label image classification which assigns at least one label to each image patch. The model achieved comparable accuracy (76.3%) to ResNet (76.4%) and EfficientNet (76.1%). Sawant and Ghosh [38] compared LinkNet backboned by MobileNetv2 with UNet for LCLU classification using Sentinel-2 images in two Indian cities; they found the lightweight LinkNet (61.3%) underperformed the UNet (71.3%). Wang et al. [39] combined 1D CNN and transformer architectures and achieved high accuracy (96.8%) using limited training samples (~1200 samples) for crop mapping with 1-year pixel time-series of Sentinel-2, although it was not compared to other models.

For lightweight models using Landsat data, Sencaki et al. [40] combined 1DCNN and a bi-directional Gated Recurrent Unit (GRU) for land cover classification in an Indonesia

city using Landsat 8 time series and the lightweight model with 20k parameters (92.2%) obtained better accuracy than ResNet (89.8%) and TempCNN (91.1%). Wan and Yong [41] proposed a lightweight CNN based on EfficientNetV2 and DeepLabV3+ for binary water classification using Landsat images. The lightweight model (95.3%) surpassed the other heavy 2D CNN models (e.g., DeepLabV3+ (93.2%), DeepwatermapV2 (94.5%), WatNet (93.0%)). Martono et al. [42] proposed a lightweight 1D DL algorithm with ~10k parameters by combining 1D CNN and bi-directional GRU for land cover classification using Landsat 8 time-series data. The model (95.8%) achieved comparable results with TempCNN (95.7%) and Long Short-Term Memory (LSTM) (95.3%). Table 1 below summarizes the studies on lightweight deep neural networks using Sentinel-2 and Landsat data, including their application, classification type and input data characteristics (spatial patches and/or time series), and comparative methods. The summary showed that studies either used CNN-based lightweight models to handle spatial features for LCLU classification, or 1DCNN/temporal models to extract temporal features, primarily for crop classification. Furthermore, most studies that compared lightweight models to their heavyweight counterparts often covered a local area and short time periods.

Table 1. Summary of studies on lightweight deep neural networks using Sentinel-2 and Landsat data.

No.	Reference	Application	Spatial Patches	Time Series	Classification Type	Compared Methods
1	Mazzia et al. (2019) [33]	Crop	No	Yes	Pixel-based	SVM, RF, XGBoost
2	Garnot and Landrieu (2020) [34]	Crop	No	Yes	Pixel-based	TAE, GRU, TempCNN, ConvLSTM, Transformer
3	Corbane et al. (2021) [35]	Built-up	Yes	No	Pixel-based	
4	Arrechea-Castillo et al. (2023) [36]	LCLU	Yes	No	Pixel-based	AlexNet, DenseNet, EfficientNet, GoogleNet, VGG, ResNet, ZFNet
5	Papoutsis et al. (2023) [37]	LCLU	Yes	No	Scene-based	ResNet, DenseNet, VGG, ViT, EfficientNet
6	Wang et al. (2024) [39]	Crop	No	Yes	Pixel-based	
7	Sawant and Ghosh (2024) [38]	LCLU	Yes	No	Pixel-based	UNet, VGG
8	Sencaki et al. (2023) [40]	LCLU	No	Yes	Pixel-based	ResNet, TempCNN, MCDCNN
9	Wan and Yong (2023) [41]	Water	Yes	No	Pixel-based	DeepLabV3+, DeepwatermapV2, WatNet
10	Martono et al. (2025) [42]	LCLU	No	Yes	Pixel-based	TempCNN, 1DCNN; Dual Bi-LSTM

Note: No. 1–7 for Sentinel-2, 8–10 for Landsat.

From Table 1 it can be seen that no study has yet proposed or applied specific lightweight models that can simultaneously manage spatial, spectral, and temporal features for medium-resolution sensor time-series data, nor have several different lightweight models been compared for their performance in multiple-class LCLU classification using medium-resolution optical data. As Landsat provides freely available, continuous long-term, global-coverage data supporting diverse applications, it is important to propose, apply, and compare lightweight models that can efficiently leverage this rich data in multiple-class LCLU classification. This study presents the first large-scale comparison of lightweight models for spatial–spectral–temporal Landsat time-series LCLU classification. By evaluating seven popular computer vision models across five parameter scales (3k–50k) over the Conterminous United States (CONUS), practical guidance on how lightweight

designs can outperform traditional CNNTransformer hybrids in accuracy and stability is provided.

To address this gap, this article aims to answer the following research questions on transitioning the existing lightweight algorithms of medium-resolution classification tasks:

1. How do different lightweight deep learning architectures perform in Landsat time-series LCLU classification?
2. Can lightweight deep learning models achieve comparable or better classification performance to a traditional CNN + transformer hybrid classifier?
3. Is there a specific model complexity that offers the highest classification improvement for a lightweight vs. traditional classifier?

2. Data

2.1. Study Area

The study area was the CONUS. This study used two datasets. One is the Land Change Monitoring, Assessment, and Projection (LCMAP) reference dataset. The LCMAP reference dataset contains ~25,000 Landsat 30 m resolution plots randomly selected across the continental U.S. Analysis Ready Data (ARD) grid system with annual labels from 1984 to 2018 [43]. The LCMAP data were a cooperation between the USGS LCMAP group and the U.S. Forest Service Landscape Change Monitoring System [44]. Here, we further enhanced that dataset by visually inspecting and manually reclassifying the LCMAP plots using Google Earth high-resolution imagery to ensure consistency with the other manually labeled block data used in this study. The LCMAP labels include seven LCLU classes (water, developed, grass/shrub, forest, bare, agriculture, and wetland) with annual time steps from 1996 to 2023 (including both the start and end year). To ensure high data quality, some plots were excluded from the dataset when the interpreters could not confidently assign a label to a plot. As a result, 24,279 high-confidence plots were retained.

For each plot, yearly Landsat data sequences with six bands (Blue, Green, Red, NIR, SWIR1, and SWIR2) were downloaded and generated using the Google Earth Engine (GEE) platform. Since GEE contains Landsat Collection 2 data, no further processing was conducted for alignment between the different sensors. Each yearly sequence is a data cube with 7 × 7-pixel image patches covering a year, with a label for the center pixel. A 7 × 7-pixel neighborhood was selected to provide sufficient spatial context for the models to extract spatial and spectral features. All data available from Landsat 5, 7, 8, and 9 sensors were incorporated in the downloaded data. For data quality, Landsat "radsat_qa" and "pixel_qa" quality bits were used for each pixel in the 7 × 7 patches to identify radiometric saturation and cloud or cloud shadow conditions (medium- or high-confidence), as well as ice/snow-covered area. Pixels identified as cloud, cloud shadow, covered by ice/snow, or saturated were masked out. This masking included the center pixel of any patch if that pixel met the masking criteria. A 7 × 7 patch was discarded entirely if all pixels were masked; otherwise, valid pixels (including the center pixel, if it was valid) were preserved. In this study, only the above-mentioned six bands of data were used rather than including supplementary data like a Digital Elevation Model (DEM), climate data (e.g., temperature, precipitation), or spectral indices (e.g., NDVI) for data simplicity and computational efficiency.

The other block data were used by Mountrakis and Heydari [45]. The data contains 84 representative image blocks of the 84 level III Environmental Protection Agency (EPA) ecoregions across CONUS. Each block covers 10 km × 10 km and is composed of 333 × 333 pixels (at 30 m spatial resolution). The labels for each pixel in the blocks were manually assigned for each year from 2000 to 2019 (including both the start and end year) according to a modified Anderson classification system. Then, the labels were visually

checked again by a team of interpreters using Google Earth high-resolution imagery. Similarly to LCMAP, the Landsat data cube of each block from 2000 to 2019 was downloaded using the GEE platform.

2.2. Sampling

For a fair and objective comparison of the models' ability, a fixed internal testing (validation) dataset and an external testing (test) data were used for all models, and 25 repetitions of the training dataset were generated for generalization and robustness. Both the LCMAP and block data were used to generate the datasets considering the diversity of spatial heterogeneity and incorporating a large sample size. For a good geographic distribution of the selected blocks for the train and test data, four geographic regions were created by merging 15 level I ecoregions over the contiguous United States: Eastern Forests, Great Plains, Deserts and Drylands, and Western Forests and Mountains. Detailed definitions of these ecoregions (areas with similar ecosystems and environmental resources) are provided by the EPA (https://www.epa.gov/eco-research/ecoregions, accessed on 1 December 2025). Figure 1 shows the locations of the 84 blocks and the four geographic regions. Table 2 shows the distribution of the blocks across the four ecoregions. The blocks selected for the train and test data cover all the four different geographic regions. Each block is assigned exclusively to either the train, internal test, or external test set to ensure spatial independence.

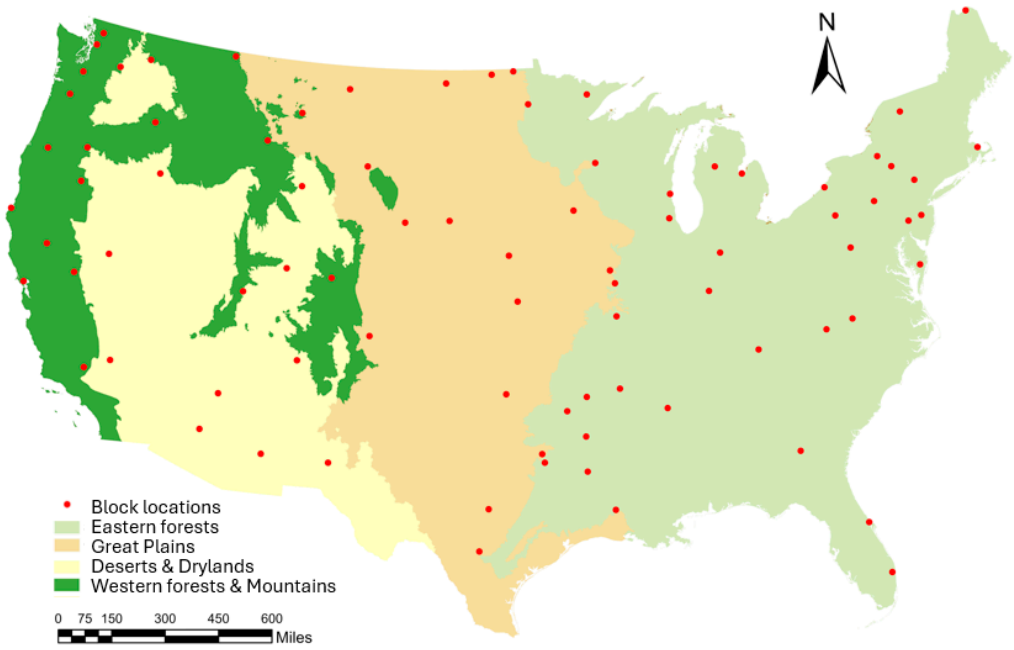

Figure 1. The 84 blocks (red circles) with four ecoregions (Eastern Forests, Great Plains, Deserts and Drylands, and Western Forests and Mountains) in the conterminous United States.

Table 2. Distribution of 84 blocks across the four ecoregions.

Metric	Eastern Forests	Great Plains	Deserts and Drylands	Western Forests and Mountains
Number of blocks	38	15	14	17
Proportion (%)	45.2	17.9	16.7	20.2

The external test data used for accuracy assessment includes 318,205 samples. Each sample consists of a yearly Landsat time series of 7 × 7-pixel image patches, with the class label assigned to the center pixel. Among these, 91,822 samples were generated from

4386 LCMAP reference plots (i.e., 30 m resolution pixels). These plots were randomly sampled from LCMAP reference data. For each reference plot, the plot location was treated as the center pixel, and 7 × 7 image patches were extracted for each year. Although a plot contains multiple years of observations, each yearly 7 × 7 patch sequence was treated as a test sample. To avoid data leakage, all yearly patch sequences generated from the same plot were assigned to the same data split. Consequently, no samples from the same plot were shared across the training, internal testing, and external testing datasets. The test data also contains 226,383 samples generated from 17,783 pixels across 12 blocks. The 12 blocks were selected by randomly sampling three blocks from each of the four ecoregions, thus ensuring spatial independence for the external validation dataset. Pixels within these blocks were drawn to approximately match the class proportion of the LCMAP reference plots, improving consistency of class representation across external test data sources. The internal test data has 44,411 LCMAP samples from 2147 plots. The plots in the internal test data were randomly sampled from the remaining LCMAP plots excluding those in the external test data. The internal test data also contains 12,743 block samples from 1001 pixels across seven blocks covering the four geographic regions.

For the training repetition data, each repetition includes a class-balanced training dataset with 500,000 training samples. Although this sample size is larger than what some small lightweight deep learning models typically require, this large training dataset minimizes sample-driven noise and ensures that algorithmic behavior, not data variability, is the primary source of performance variation. Balanced training data was used to eliminate class bias and improve generalization. The training data size is a trade-off between data volume, data diversity and generalization considering the LCMAP and block data. With seven classes, each class has about 71,429 samples. For one repetition of the training dataset, first, 15,000 LCMAP plots were randomly selected from the rest of the LCMAP plots excluding the internal and external testing data. Then, the training dataset was balanced to 71,429 samples per class by either adding block samples or removing surplus LCMAP plots. The block samples were selected from a pool of 24 selected blocks. The 24 blocks were selected by randomly sampling six blocks from each of the four ecoregions from all remaining blocks, specifically excluding those selected in the internal and external testing dataset. This procedure was repeated 25 times to generate 25 training repetitions.

3. Methods

3.1. Models

Seven lightweight DL models and one traditional CNNTransformer hybrid as the benchmark were evaluated. These models were proposed and selected to represent diverse lightweight DL architectures that can be applied to RS time-series classification. These models can be categorized into four groups depending on the underlying architecture type: convolutional + recurrent, convolutional + transformer, 3D convolutional, and video transformer architectures. A brief overview of each model is provided below.

3.1.1. Lightweight Models

Convolutional and Recurrent Models

MobileNetSRU is a hybrid architecture combining MobileNetV3-Small [11] for spatial feature extraction and Simple Recurrent Units (SRUs) [46] for temporal feature extraction. The input data cube was first processed by MobileNetV3-Small blocks, then the extracted spatial features adding the center pixel spectral features were passed to SRUs for further temporal feature extraction and then a classification head was used to generate the final class labels. MobileNetV3-Small represented a new generation of efficient convolutional networks (i.e., MobileNets) optimized for mobile and embedded devices. It was developed

through NAS using the NetAdapt algorithm and refined with efficient design components [47,48]. MobileNetV3 variants are widely used in real-time vison tasks such as detection [49]. They are also used in RS, such as in efficient sugarcane mapping [50]. An SRU is a lightweight recurrent unit designed for high parallelization and computational efficiency. SRUs remove the dependency between the time steps of traditional RNNs (like LSTMs), allowing the heavy matrix multiplications for all time steps to be processed in parallel. SRUs are used in the Natural Language Processing (NLP) field, such as for text classification and question answering [46]. SRUs were also applied in automatic speech-recognition systems [51].

ConvNextKanSRU follows a similar hybrid design, combining ConvNextKan for spatial processing and SRUs for temporal processing. ConvNextKan [52] integrates the ConvNext architecture [53] with the Kolmogorov–Arnold Network (KAN) [54]. ConvNext modernizes traditional CNNs with design elements inspired by ViT [55]. ConvNext models were widely applied in computer vision tasks. For example, ConvNext models were employed for semantic segmentation and object detection [56,57]. The KAN introduces learnable activation functions applied to weights, inspired by the Kolmogorov–Arnold representation theorem, allowing compact architectures to achieve performance comparable to larger multilayer perceptrons (MLPs) [54]. KAN has shown promising results in scientific discovery [54]. KAN was also used in hyperspectral image classification and achieved comparable results to MLP [58].

Convolutional and Transformer Models

MobileNetTransformer combines MobileNetV3-Small for spatial feature extraction with an efficient transformer module that employs linear attention instead of traditional quadratic self-attention for temporal feature extraction. The linear-attention mechanism modifies the attention calculation to reduce this complexity to linear scaling [59]. This model was primarily used in the NLP field, such as in long-context language modeling [29]. Specifically, it was employed in Large Language Models (LLMs) for recall-intensive tasks such as question answering on long documents [59].

ConvNextKanTransformer combines ConvNextKan [52] with the same efficient transformer module [59]. ConvNextKan captures spatial information, while the transformer extracts temporal features, similar to the MobileNetTransformer architecture.

3D Convolutional Models

MoViNet (Mobile Video Network) is an efficient 3D convolutional network designed for real-time video understanding on resource-constrained mobile and edge devices [60]. It incorporates several efficient techniques to balance accuracy and computational efficiency, achieving state-of-the-art results on standard video recognition benchmarks. MoViNet has been successfully applied to action recognition tasks, enabling edge devices to classify human activities in live video streams [60].

TVN (Tiny Video Network) is another compact convolutional video model optimized for fast and effective video recognition tasks [61]. TVN adopts a simplified, decoupled approach to spatiotemporal learning, using a combination of 2D CNNs for spatial feature extraction and 1D CNNs for temporal relationship modeling. TVN models were effectively used in activity monitoring and surveillance systems for near-real-time classification of human actions [61].

Video Transformer Models

EVT (Efficient Video Transformer) is designed for efficient video recognition and understanding [12]. It employs a dynamic spatial–temporal token-selection mechanism by selecting only the most informative spatial and temporal tokens to reduce computational

cost while maintaining accuracy. EVT models were used in action recognition tasks to classify human actions in videos, while achieving comparable performance to state-of-the-art video transformers [12].

To the best of our knowledge, these seven lightweight DL models are for the first time applied to RS time-series classification.

3.1.2. Benchmark

CNNTransformer, the benchmark model used in this study, combines a traditional CNN with the standard transformer architecture proposed by Vaswani et al. [9]. This model represents a conventional CNN and transformer hybrid, allowing comparison between lightweight and traditional architectures. The transformer, a popular DNN architecture, revolutionizes sequence-to-sequence modeling by relying entirely on the self-attention mechanism. It can weigh the importance of different parts of the input sequence relative to the current token, allowing it to capture long-range dependencies efficiently. It can also enable parallel processing for faster training times. This transformer became a foundational building block for DL tasks, primarily in NLP (e.g., machine translation, text generation) and computer vision (e.g., image classification, object detection) [55,62].

3.2. Model Training

Model training consisted of two steps: architecture selection and hyperparameter selection. First, architecture selection defined the architectures and parameter scales for all models. To systematically investigate how the model performance changes with model size, and to allow a fair comparison across architectures with different model complexities, each model was implemented with five parameter scales (approximately 3k, 5k, 10k, 25k and 50k parameters), representing very small, small, medium–small, medium, and large variants. The model scales (3k to 50k parameters) were defined based on the training set of 500,000 samples, following the "rule of ten" to prevent overfitting by keeping the training-sample-to-model-parameter ratio at least 10:1. Initially, 5k, 25k, and 50k scales were tested. Following preliminary results with comparable performance between the 25k and 50k scales, 3k and 10k scales were added to better observe performance changes in the small-to-medium models. These variants are created by searching architectures from the model parameter space as shown in Table 3. Each parameter scale of a model had multiple representative architectures (e.g., 7 to 10 architectures per model size) for subsequent hyperparameter selection.

Table 3. Model-specific parameter ranges for the eight models.

Model	Parameters	Ranges
MobileNetSRU	MobileNet layers	2, 3, 4, 5, 6
	Convolution output channels	4 to 80 (step 4)
	SRU layers	1, 2, 3, 4, 5, 6
	Hidden feature size	4 to 120 (step 4)
ConvNextKanSRU	ConvNext blocks	1, 2
	Convolution output channels	4 to 80 (step 4)
	KAN output features	2 to 18 (step 2)
	SRU layers	1, 2, 3, 4, 5, 6
	Hidden feature size	4 to 120 (step 4)

Table 3. *Cont.*

Model	Parameters	Ranges
MobileNetTransformer	MobileNet layers	2, 3, 4
	Convolution output channels	4 to 80 (step 4)
	Transformer layers	1, 2, 3, 4
	Embedding dimension	25 to 120 (step 5)
	Attention heads	1, 2, 4, 8
ConvNextKanTransformer	ConvNext blocks	1, 2
	Convolution output channels	4 to 80 (step 4)
	KAN output features	2 to 18 (step 2)
	Transformer layers	1, 2, 3, 4
	Embedding dimension	25 to 120 (step 5)
	Attention heads	1, 2, 4, 8
MoviNet	3D CNN blocks	1, 2, 3
	Convolution channels	8 to 80 (step 4)
	3D convolution kernel size	(3, 3, 3), (1, 3, 3), (5, 3, 3)
	3D stride	(1, 1, 1), (1, 2, 2)
	3D padding	(1, 1, 1), (0, 1, 1)
	Head MLP hidden size	8 to 128, step 8
TVN	Spatial convolution (kernel size, stride)	(3, 1), (3, 2), (5, 1), (5, 2)
	Temporal pooling (kernel size, stride)	(2, 2), (2, 3), (3, 2), (4, 2)
	Context gating convolution (kernel size, stride)	(3, 1), (3, 2), (1, 1)
	CNN output channels	10 to 52 (step 4)
	Head MLP hidden size	12 to 72 (step 4)
EVT	Transformer blocks	1, 2, 3, 4
	Attention heads	1, 2, 4
	Embedding dimension	2 to 16 (step 2)
	MLP ratio	0.5 to 3.0 (step 0.5)
CNNTransformer	Convolution layers	1, 2, 3
	CNN channels	8 to 40 (step 4)
	Transformer layers	1, 2, 3
	Embedding dimension	8 to 80 (step 4)
	Attention heads	2 to 8 (step 2)
	Feedforward network dimension	8 to 160 (step 8)

Second, hyperparameter selection aimed to identify the optimal training configuration for each architecture. Jointly searching all combinations would impose extremely high computational cost due to high architectural and hyperparameter complexity [63]. To control computational cost while ensuring fair comparison, a coarse-to-fine grid search was adopted. This approach avoids the high computational overhead and stochastic noise of automated search, ensuring that performance differences reflect architectural characteristics rather than unequal optimization budgets. A coarse search with wide parameter intervals was used to identify general regions of good performance, followed by a fine search with narrower intervals around these regions to determine the optimal configuration. For each parameter scale, all candidate architectures were evaluated under the coarse search, and the top two performing architectures were refined through fine-grid search to identify the optimal configuration of the architecture and hyperparameters. This staged search strategy substantially decreases computational cost, while still enabling robust exploration of the hyperparameter space. Table 4 summarizes the hyperparameter and search ranges in this process.

Table 4. Hyperparameter values and ranges.

Hyperparameter	Values/Range
Learning rate	1×10^{-4} to 1×10^{-2}
Weight decay	1×10^{-6} to 1×10^{-2}
Learning rate scheduler	Cosine, one cycle
Optimizer	Adam, Adamw
Loss function	Categorical cross-entropy
Epochs	100
Early-stopping epochs	10
Batch size	64,128
Drop out	0.0 to 0.5

4. Results

To evaluate model stability and robustness, the optimal architecture and hyperparameter configuration was then retrained across the 25 training repetitions. These repetitions were conducted using the same fixed architecture and hyperparameter settings but employed different training data samples for each repetition. Using a single repetition for model selection prevents information leakage and retraining the selected configuration on the remaining repetitions with fixed test sets isolates the effects of training data variation, enabling fair assessment of model generalization and stability. To further assess model robustness under limited-data scenarios, we performed an additional experiment using a small training dataset consisting of 50k samples. Given the higher variance expected in limited-data conditions, models were evaluated over 50 training repetitions while maintaining the same model selection and testing protocol. In this section, we first present the benchmark model performance, followed by the performance of lightweight models relative to the benchmark, a further investigation of the optimal lightweight model performance relative to the benchmark, and model performance with small training dataset.

4.1. Benchmark Performance

As mentioned, the CNNTransformer hybrid was the benchmark model used in this study for comparison with the lightweight models for Landsat time-series LCLU classification. Figure 2 presents the benchmark performance (mean F1 and per-class F1) across 25 training repetitions. The mean F1 distribution, typically falling between 72 and 75% depending on model size, showed high stability, with moderate variability across repetitions, indicating that the benchmark is robust to variation in training data. The mean F1 increased with model size from 3k to 25k, reflecting the expected benefits of wider attention layers and deeper temporal modeling capacity. However, the 25k and 50k models achieved nearly identical mean F1, indicating that model capacity reached a point of saturation, where additional parameters offered limited marginal gains. The tighter boxplots at 50k further supported this interpretation. While the mean performance did not improve beyond the 25k model, the narrower spread indicated that the 50k model showed lower variability across training repetitions, meaning more stable optimization and more consistent convergence. Larger models often have smoother loss landscapes and more redundant capacity, thus reducing run-to-run fluctuations even when accuracy plateaus.

Class performance revealed clear differences among the LCLU types. Some land cover types (e.g., water and agriculture) displayed a consistently high F1, while more dynamic classes (e.g., wetland or grass/shrub) exhibited greater variability.

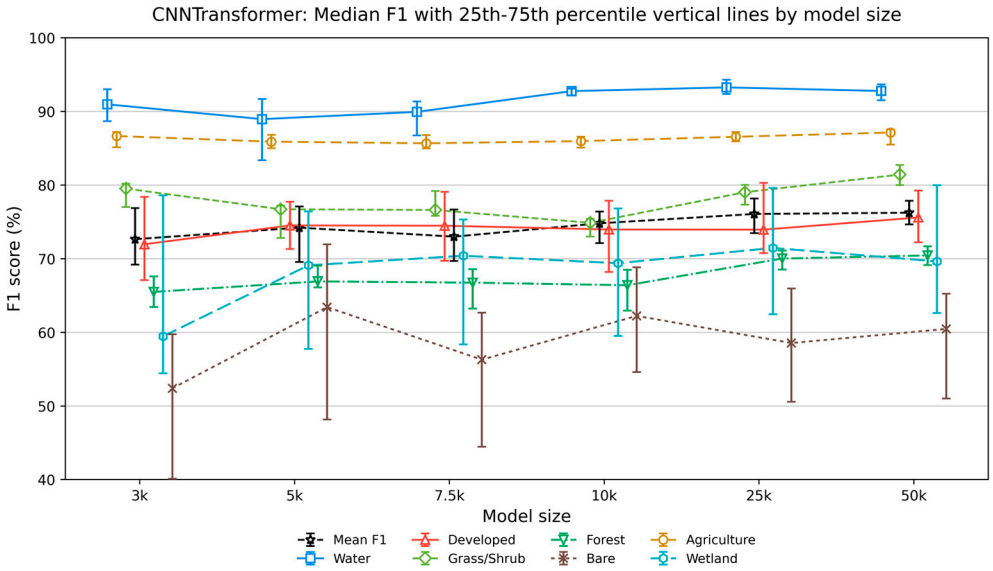

Figure 2. Benchmark (CNNTransformer) mean and per-class F1-score line plot by model size from 25 training repetitions.

The water and agriculture classes consistently achieved the highest F1 scores (85–90%), reflecting the water class's strong spectral and temporal separability and low within-class heterogeneity and the agriculture class's strong spatial and temporal separability. The developed and grass/shrub classes followed, with relatively stable performance (approximately 75–80%) across repetitions. Developed areas had spatial contrast (e.g., edges, boundaries) and stable reflectance with weak seasonality, while grass/shrub areas had strong but consistent seasonal patterns (i.e., large spectral amplitude with consistent annual phenological timing). Forest and wetland exhibited notably lower F1 values (roughly 65–70%); however, forest showed lower variability while wetland displayed substantially higher variability, indicating its more inconsistent separability across training repetitions. Forest areas tended to have homogeneous canopy structures and stable phenological cycles (e.g., spring–summer–fall temporal changes) across years. However, wetlands often exhibited mixed vegetation and water signals within Landsat pixels at 30 m resolution, and wetlands also usually included inconsistent temporal changes across years, such as affection of flooding and drought. Furthermore, the spectral similarity between wetlands and adjacent land cover types increases class ambiguity in optical imagery, potentially introducing greater label uncertainty compared to more spectrally homogeneous classes. Bare areas had the lowest F1 values (52–62%), reflecting their spectral ambiguity and weak temporal signal. Bare land surfaces resembled the developed, grass/shrub, and even agriculture classes depending on soil moisture, mineral content, or tillage. They often lacked consistent phenological behavior, limiting the extraction of temporal patterns. The bare land class also had large intra-class variability, as it can include deserts, mine excavations, riverbanks, and burn scars, each with very different spectral patterns.

Overall, the benchmark's performance provided a reliable baseline for evaluating whether lightweight models can approximate or exceed the traditional CNNTransformer hybrid.

4.2. Performance of Lightweight Models Relative to the Benchmark

To evaluate performance consistency across different model architectures and parameter scales, a Two-Way ANOVA with blocking was conducted as shown in Table 5. The analysis revealed significant main effects for both the model ($F(7, 936) = 12.12$, $p < 0.001$) and parameter scales ($F(4, 936) = 5.87$, $p < 0.001$). The blocking factor (training repetition) was also highly significant ($p < 0.001$), confirming that the performance varied naturally

between different training repetitions. Most notably, a highly significant interaction effect between the model and parameter scales was observed ($F(28, 936) = 134.49$, $p < 0.001$). This interaction indicates that the performance gain from increasing the parameter size is dependent on the specific model. A more detailed analysis of impact of model size separated by model type is needed.

Table 5. ANOVA results for mean F1 scores across models and model sizes from 25 training repetitions.

Source	SS	df	MS	F	*p*-Value
Model	0.093	7	0.013	12.12	<0.001
Params	0.026	4	0.007	5.87	<0.001
Block	0.678	24	0.028	25.77	<0.001
Model × Params	4.127	28	0.147	134.49	<0.001
Residual	1.026	936	0.001	-	-

Note: SS = Sum of squares; df = degrees of Freedom; MS = mean square; Params = parameter scales.

To further evaluate if lightweight models can perform comparably to or better than the benchmark, the mean F1 difference was calculated by subtracting the benchmark's mean F1 score from the model's mean F1 score for the same training repetition. Figure 3 summarizes each lightweight model's mean F1 difference over the benchmark across model sizes (3k–50k). A positive value indicates performance surpassing the benchmark and vice versa. For better visualization, EVT values at small model sizes (3k and 5k) were removed due to poor performance (lower than −20%) relative to the benchmark.

The CNNRNN models (MobileNetSRU and ConvNextKanSRU) achieved the strongest performance across all model sizes, consistently outperforming the benchmark. MobileNetSRU exceeded the benchmark at every model size, including at smallest size (3k parameters), where it outperformed the benchmark by a substantial margin, demonstrating exceptional parameter efficiency and stable temporal modeling [11,46] via SRU (more details in Section 4.3). ConvNextKanSRU followed closely but was typically slightly below MobileNetSRU in performance at most model sizes; however, at larger model sizes, ConvNextKanSRU achieved comparable performance to MobileNetSRU. At 25k, the highest mean F1 surpassed MobileNetSRU, suggesting that the ConvNext + KAN spatial encoder benefited more from increased model size, allowing the architecture to realize its full spatial and temporal modeling potential at a larger model size. The variability of the two models decreased as model size increased, highlighting improved model stability and reduced sensitivity to training data variation. This pattern indicated that larger lightweight models gain stronger representational capacity, allowing them to learn more consistent spatial and temporal features across repetitions. As parameter budgets increase, both architectures become less prone to overfitting individual training draws and better capture class-invariant patterns, resulting in tighter performance distributions and more reliable generalization.

The lightweight CNNTransformer models (MobileNetTransformer and ConvNextKanTransformer) showed a model-size-dependent pattern. Transformer-based lightweight models underperformed the benchmark at a small model size (3k) but became competitive at larger model sizes. At 3k, both models fell below the benchmark (negative F1 difference), showing instability and underfitting. The two models scaled down less efficiently than the benchmark as they carry larger structural overhead in their MobileNet and ConvNextKan backbones (e.g., depthwise convolutions, and squeeze-and-excite, normalization, and residual connections) and attention mechanisms (e.g., projection matrices), leaving insufficient effective capacity for learning meaningful spatial–temporal features at this extremely small size. As the model size increased (5k to 25k), the relative performance of

MobileNetTransformer improved, with F1 difference centers above the benchmark. Between the 5k and 25k model sizes, the parameter numbers became adequate to support both spatial and temporal modeling. MobileNet and ConvNextKan started to build meaningful spatial representations and the linear attention became expressive enough to capture annual phenology.

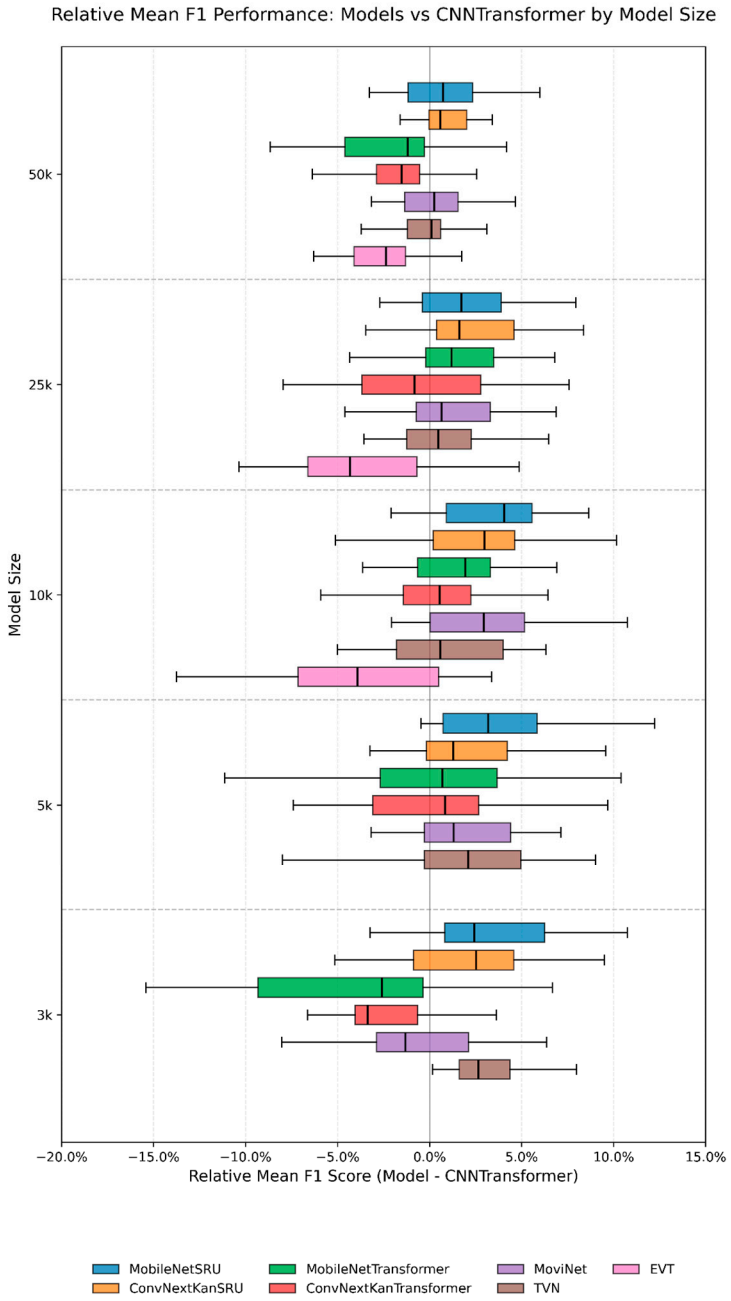

Figure 3. Models relative to benchmark mean F1 difference boxplots from 25 training repetitions. Positive values indicate performance surpassing the benchmark; negative values indicate performance falling below the benchmark.

Then, relative performance decreased with a model size increase from 25k to 50k, indicating lightweight transformers could not achieve comparable performance to traditional transformers for large model sizes. Both models fell below the benchmark again at 50k, suggesting that the linear attention mechanism, while efficient for small models, became less effective than the traditional quadratic self-attention as model size increased. At large model sizes, it is possible that linear attention could not exploit the increased capacity for

the following reasons: it compressed temporal interactions too aggressively and cannot model fine-grained pairwise relationships between time steps as traditional transformers do by design [59]. Therefore, the extra parameters could not improve temporal modeling quality. In contrast, traditional self-attention used full pairwise comparisons between time steps, enabling larger models to exploit their representational power to utilize the additional capacity better [9]. This decrease may also occur in lightweight CNNs, but this is not likely. For example, MobileNets were specifically designed to be highly parameter-efficient by using tricks such as depthwise separable convolutions to achieve comparable or higher accuracy with a much smaller parameter size than heavy CNN models [46,48]. If traditional CNNs are forced to have the same parameter size as MobileNets, the traditional CNNs' accuracy often drops as they cannot leverage a small parameter size as efficiently as MobileNets. Although there is no direct comparison example found in LCLU classification, Melyani et al. [64] found MobileNetV2 achieved better accuracy (5% gain) than the traditional CNN of DenseNet 121 at the same parameter scale in eye disease image classification. The variability of MobileNetTransformer decreased as model size increased to 25k, then increased from 25k to 50k; model size also indicated model-size-dependent patterns. However, the non-monotonic stability observed in ConvNeXtKanTransformer (3k more stable than 5k) likely reflected optimization dynamics interacting with the KAN component, where certain intermediate capacities lead to mismatch between attention head size and spatial feature dimensionality. Temporal transformer modeling requires larger representational capacity due to reliance on attention mechanisms and weaker inductive bias [9], and transformers often require large datasets (e.g., millions of samples) to reach their full potential, because they are essentially trying to learn the entire sequential structure from the data, which requires vast amounts of evidence.

The 3D CNN models (MoviNet and TVN) achieved comparable or better performance than the benchmark at all model sizes. At 3k, MoviNet had mean F1 difference center below 0%, showing slightly weaker average performance than the benchmark; however, at other model sizes, MoviNet became competitive, with centers exceeding the benchmark. Moreover, the relative improvement to benchmark first increased (5k to 10k) then decreased (25k to 50k) as model size increased. This underperformance indicated the inefficiency of 3D CNN compared to CNNTransformer at extremely small parameter sizes. MoViNet used 3D kernels (e.g., $3 \times 3 \times 3$ or $1 \times 3 \times 3$) to capture spatial and temporal correlations simultaneously and the model design mechanisms (e.g., depthwise 3D convolutions, temporal buffering) only begin to function effectively once the channel width is high enough. In contrast, the CNNTransformer hybrid separated spatial and temporal modeling, and this separation allows the CNNTransformer to remain functional even at 3k, whereas MoViNet became too shallow and narrow to extract meaningful 3D features. This was also indicated by the hybrid design of TVN with 2D + 1DCNN.

As model size increased from 5k to 10k, MoviNet started to obtain enough capacity to capture joint spatial–temporal features and gain better relative performance compared to the benchmark. However, the relative improvements of MoviNet declined at 25k and 50k, illustrating that the 3D CNN model overfitted more easily at larger model sizes. MoviNet mixed spatial and temporal features too early, which is inherent to the design of 3D CNN architectures and may amplify noise and temporal irregularity as model size increases.

TVN consistently achieved mean F1 difference values near or above the benchmark across model sizes. Particularly at 3k, TVN's relative performance was strong and stable, outperforming the benchmark and indicating that the 2D + 1D CNN architecture is more parameter-efficient than the 3DCNN architecture of MoviNet at small model sizes. However, the 3D CNN architecture of MoViNet became slightly superior to TVN at larger sizes (10k, 25k and 50k), possibly related to its joint spatial–temporal convolutions, which can

exploit model capacity to learn richer and more holistic spatiotemporal representations than the separated 2D + 1D CNN design.

The video transformer (EVT) showed substantial underperformance compared to the benchmark across all model sizes, failing to reach the benchmark's level of performance at any model size. Although performance improved with model size, EVT remained below the benchmark at large model size of 50k. The high variance also indicated unstable convergence at small and medium model sizes. Video transformers typically require large model widths and deep hierarchies to stabilize attention across space–time tokens [65,66]. EVT likely struggled to maintain consistent attention patterns at small model sizes, resulting in low relative performance and high instability.

Overall, MobileNetSRU was the best across the seven lightweight architectures. The next section compares it with the benchmark based on per-class performance.

4.3. Further Investigation of Optimal Lightweight Model Relative to Benchmark

To compare MobilenetSRU, the best-performing model overall, with the benchmark in more detail and to find where accuracy gains are maximized, two model sizes (7.5k and 15k) were added. Figure 4 shows the mean F1 and per-class F1 of MobileNetSRU relative to the benchmark across model sizes (3k to 50k).

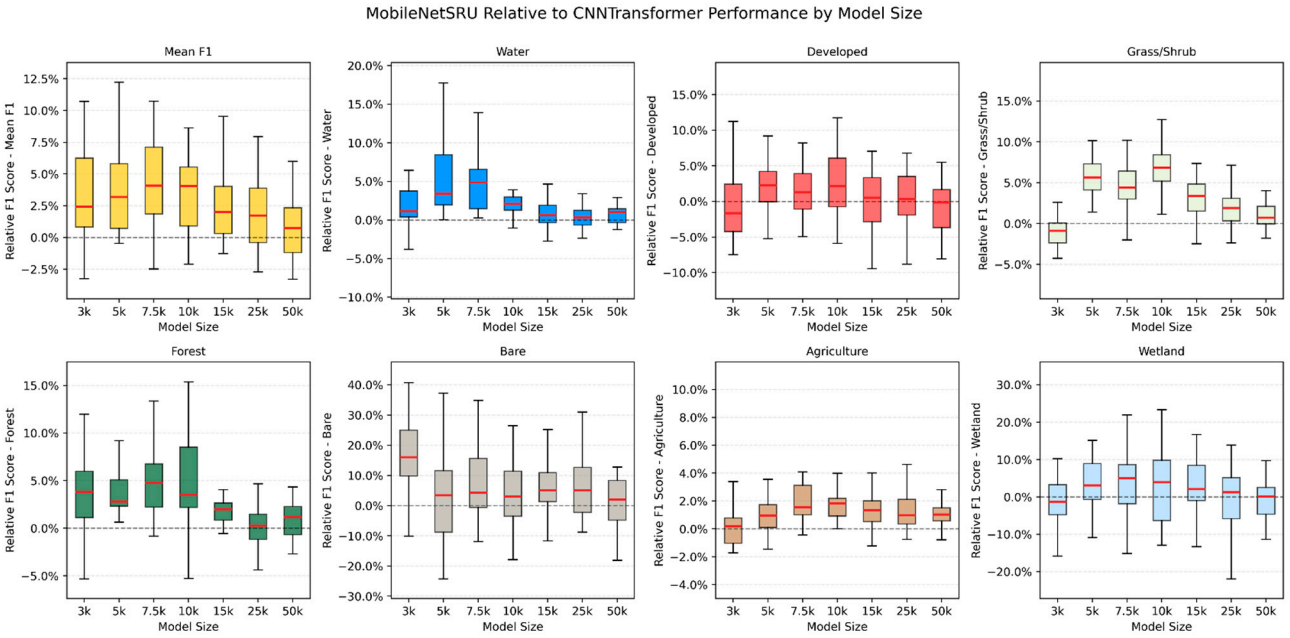

Figure 4. MobileNetSRU relative to benchmark (CNNTransformer) mean and per-class F1 score across model sizes (3k to 50k) from 25 training repetitions.

Principally, MobileNetSRU exhibited a size-dependent performance pattern, with stronger relative improvements at smaller model sizes and diminishing gains at larger sizes. This trend highlighted the specific efficiency advantages of MobileNetSRU's lightweight design and the limitations of hybrid CNNRNN architectures as model size increases. The relative improvement of MobileNetSRU was highest at 7.5k and then the performance gains declined gradually from 10k to 50k, with the 50k model showing marginal relative improvement.

At the smallest model sizes (3k–5k), MobileNetSRU also outperformed the benchmark, but with larger variability. The improvement from 3k to 7.5k reflected the range wherein MobileNetSRU obtained enough representational capacity to express MobileNet's parameter-efficient design by compact convolutional blocks and recurrent temporal mod-

eling. MobileNetV3Small effectively captured spatial information with inverted residual blocks, linear bottlenecks, and squeeze-and-excite mechanisms. Unlike transformers, the SRU avoided quadratic attention costs by providing efficient temporal modeling with simplified sequential recurrence. Furthermore, the SRU benefited from the inductive bias inherent to sequential architectures (the model did not need to use its limited parameters to learn sequential relationships from scratch) at smaller model sizes. When the model was small, these design advantages yielded better performance than the CNNTransformer benchmark. As the model size increased, however, the benchmark benefited more from the additional parameters, especially the transformer self-attention scaling with model size (e.g., wider keys/queries stabilized attention, deeper temporal receptive fields enabled modeling of fine-grained phenological differences). MobileNetSRU, in contrast, did not scale as effectively with model size because the SRU uses fixed projections (i.e., projecting the input data into gate and update components by one large linear transformation) and cannot expand temporal modeling complexity like transformers due to the SRU's sequential recurrence design.

The following analysis examines the per-class performance of MobileNetSRU relative to the benchmark models. The water class showed strong positive gains at 3k to 7.5k, gradually decreasing toward zero at larger sizes. Water has simple and stable spectral–temporal signatures. MobileNetSRU's small, efficient filters capture these reliably, whereas CNNTransformer is inefficient at small model sizes. As the model size increased, the transformer became better at modeling subtle temporal differences (e.g., seasonal changes), reducing MobileNetSRU's advantage.

The accuracy gains for the developed class peaked at around 7.5k to 10k and narrowed slightly with larger sizes. Developed areas often exhibit mixed spectral characteristics but relatively stable temporal features. Hybrid CNNRNN architectures captured texture-like patterns efficiently at small sizes. At larger sizes, the transformer layers better captured structural heterogeneity, narrowing the performance gap.

The grass/shrub class showed consistent positive gains across 5k to 15k, peaking at 10k. This class is moderately separable but temporally variable. SRU's efficient recurrence is effective for capturing moderate temporal variability but does not scale as strongly as transformer attention for long-range seasonal modeling. This explained the decline in relative performance after 15k. Notably, at the smallest model size (3k), MobileNetSRU underperformed against the CNNTransformer; this could be because both the MobileNet backbone and SRU became too compressed, while the transformer, although minimal, can still model some long-range temporal relationships through its attention mechanism, which is important for the grass/shrub class.

The forest class had positive gains at smaller sizes (3k to 10k), and relatively small gains at larger sizes, becoming near-zero at 25k to 50k. The forest class exhibited stable phenology with subtle but important temporal cues. CNNTransformer benefited from self-attention for modeling these long-term seasonal dynamics, causing MobileNetSRU to lose its advantage when the model size increased.

The bare land class showed the largest gains at small sizes (~20% improvement at 3k–10k) but this declined as model size increased (25k–50k). Bare areas had relatively simple spectral and temporal structures. CNNTransformer is inefficient under small model sizes, while MobileNetSRU captures these patterns efficiently. However, as model size increased, CNNTransformer scaled more effectively, enabling richer representation of the high intra-class variability of the bare class and reducing MobileNetSRU's relative performance advantage.

The agriculture class exhibited modest gains (~2%), first increasing from 3k to 7.5k and then decreasing at larger sizes (15k–50k). MobileNetSRU effectively captured the mid-range

phenological cycles of agriculture and performed comparably to CNNTransformer at very small sizes, reaching peak performance around 7.5k. Although this performance advantage decreased with model size, the temporal complexity of agriculture is not sufficient for the transformer to close the gap entirely, resulting in MobileNetSRU maintaining a slight performance edge even at 50k.

The wetland class showed moderate gains (5–10%) across most sizes, with gains increasing at smaller sizes (5k–7k), but a gradual decline was found as model size increased. Wetlands exhibit irregular and class-specific phenological patterns. MobileNetSRU benefits early from efficient temporal modeling, but as model size increased, the CNNTransformer better leveraged spatial–temporal heterogeneity, reducing MobileNetSRU's relative advantage and yielding a comparable performance at larger model sizes.

Collectively, the relative performance patterns indicate that MobileNetSRU was most competitive for small-to-mid-size models (5k–15k) where its architectural efficiency was maximized. The optimal point was 7.5k, after which performance declined relative to the CNNTransformer benchmark. The transformer architecture scaled more effectively with model size, providing stronger long-range temporal modeling for complex classes (forest, agriculture, and wetland). MobileNetSRU excelled at simple or moderately variable classes (water, bare land), especially with small model sizes (3k–10k). This reinforced that different architectures perform best at different model sizes, and that the lightweight CNNRNN hybrid can outperform a transformer-based hybrid when the parameter size needs to be small.

4.4. Model Performance with a Small Training Dataset

The results reported in Sections 4.1–4.3 used a substantial training dataset size of approximately 500k samples. The intent of the large reference dataset was to isolate improvements to algorithmic architectures instead of sampling limitations. Here, the analysis is expanded using a substantially smaller reference dataset of 50k samples to assess algorithmic performance under more realistic sampling sizes.

4.4.1. Benchmark Performance with 50k Training Samples

Figure 5 shows the CNNTransformer benchmark's performance on 50k training samples across 50 repetitions and model sizes ranging from 3k to 10k parameters. Compared to the 500k training sample, the benchmark exhibited a clear reduction in mean F1 scores and increased variability, highlighting the stronger sensitivity of the transformer-based model to limited training data. The mean F1 distribution typically fell between 54% and 58%, a drop of approximately 20% to 25% compared to the larger dataset. The median F1 increased from 3k to 5k parameters, remaining approximately unchanged at 7.5k, and then decreased at 10k. This non-monotonic behavior suggests that, under limited training data, increasing transformer capacity can lead to overfitting rather than improved generalization.

Class performance largely mirrored that observed in the large 500k training sample experiment but with amplified disparities. The water and agriculture classes again achieved the highest F1 scores, reflecting their strong spectral and phenological separability even under data constraints. The developed, grass/shrub and forest classes showed moderate performance with increased variance, while the wetland and bare land classes experienced notable declines in both accuracy and stability. Wetland exhibited high variability, suggesting that limited training samples were insufficient to capture its heterogeneous and temporally inconsistent signatures. The bare class displayed very low F1 scores and high variability, reinforcing its dependence on larger training datasets to disentangle spectral ambiguity and intra-class variability.

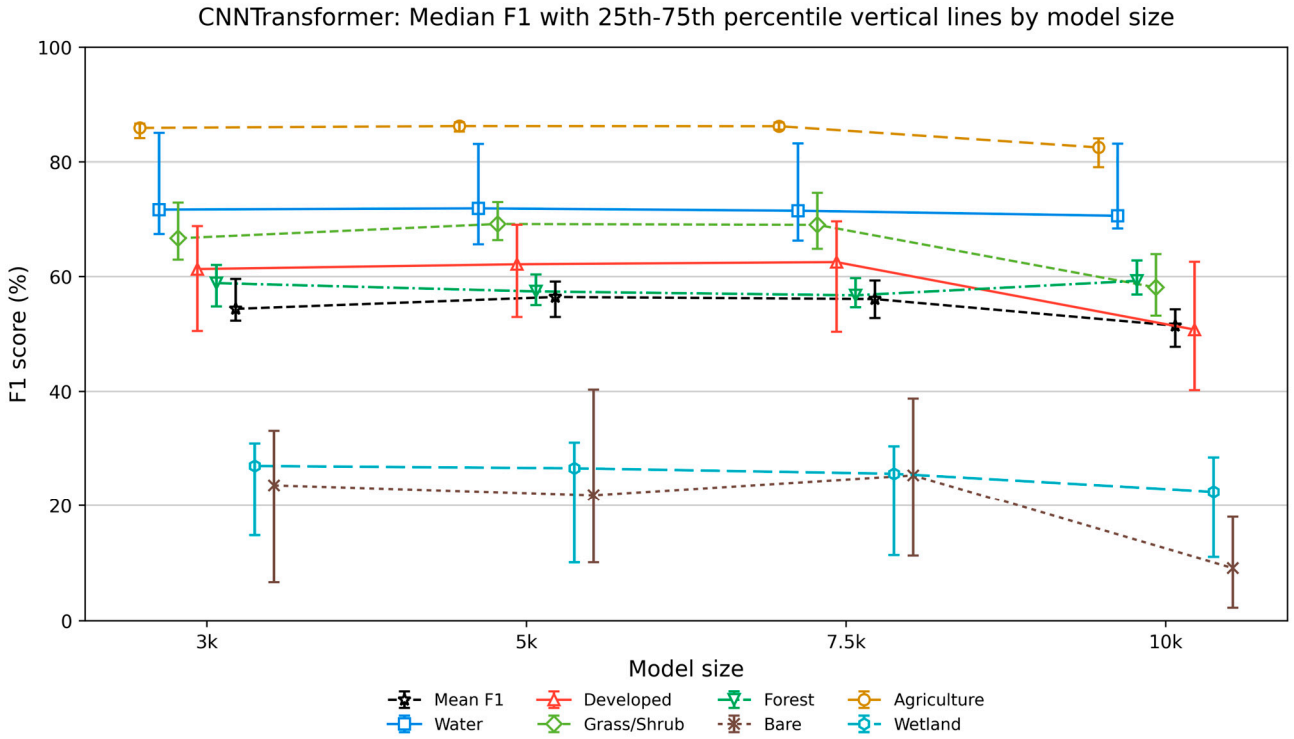

Figure 5. Benchmark (CNNTransformer) mean and per-class F1-score line plot by model size on 50k training samples from 50 training repetitions.

Overall, the benchmark results on the 50k training sample dataset demonstrate that the CNNTransformer remains functional under reduced-data conditions but suffers from degraded performance and stability.

4.4.2. Testing of Lightweight Models Relative to the Benchmark with 50k Training Samples

Figure 6 summarizes the mean F1 difference in four lightweight models (MobileNetSRU, ConvNextKanSRU, MoviNet, and TVN) over the CNNTransformer benchmark under the 50k training sample size. MobileNetSRU emerged as the most consistently robust architecture for the small dataset size, maintaining a positive performance margin over the benchmark across the 3k to 10k parameter scales. At the 3k and 5k scales, it achieved a median F1 improvement of approximately 2% to 3%, which expanded to over 4% at the 7.5k and 10k scales. This sustained superiority suggests that the SRU's recurrent gating mechanism captures temporal phenological patterns more effectively than the benchmark's attention mechanism when training samples are sparse, likely due to MobileNetSRU's ability to leverage sequential priors (temporal continuity and order dependence), maintaining both parameter and data efficiency. ConvNextKanSRU initially underperformed against the benchmark at the 3k and 5k parameter scales, with a median F1 difference of approximately -4%. At 10k, however, its relative performance improved substantially, reaching a median of 4% above the benchmark. This performance recovery is driven in part by a degradation in benchmark performance at 10k under limited training data, while ConvNextKanSRU exhibits comparatively more stable behavior. This suggests that while the ConvNextKanSRU architecture may be disadvantaged under constrained parameter and data combinations, it becomes more resilient as model capacity increases.

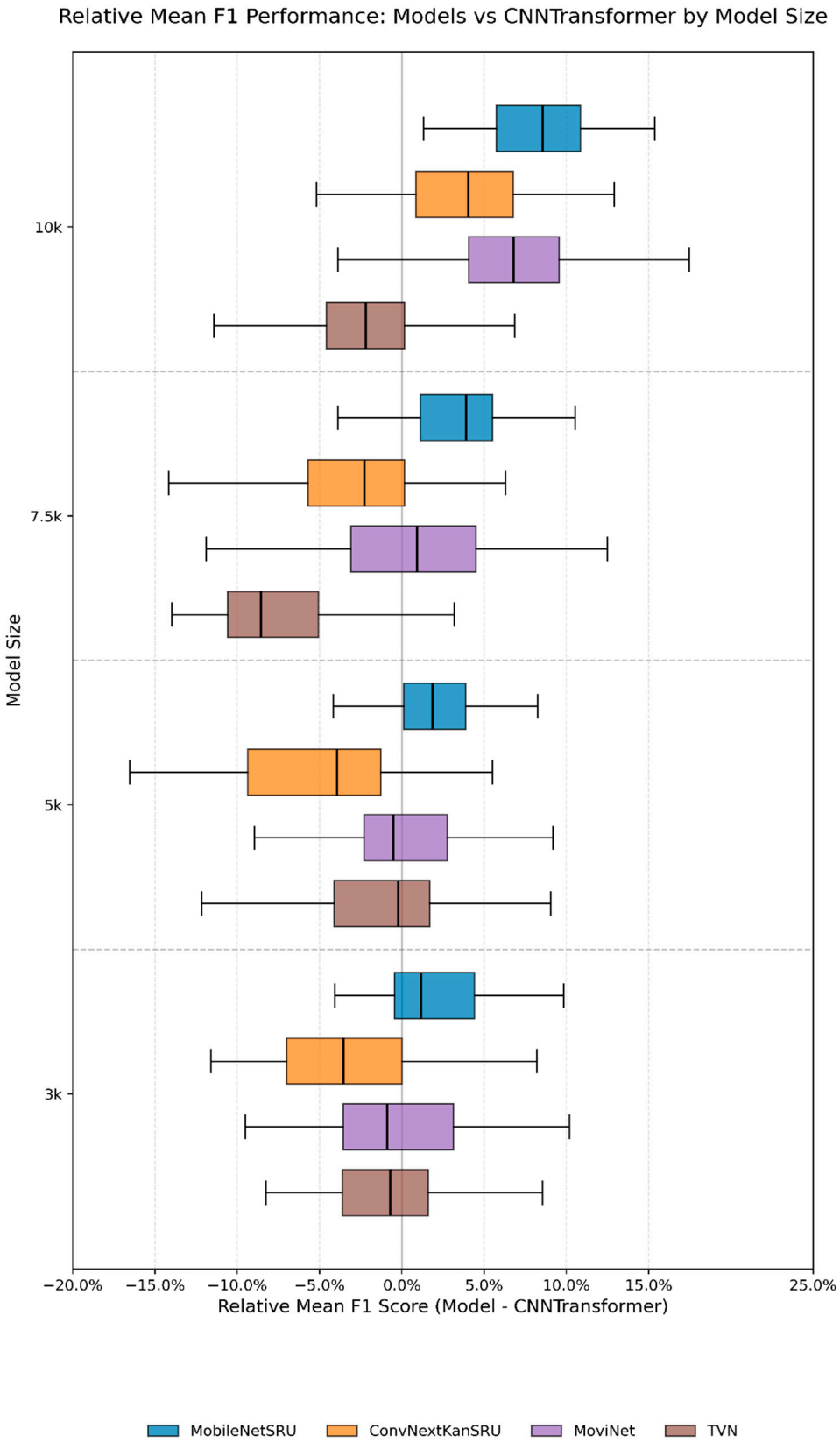

Figure 6. Models relative to benchmark mean F1 difference boxplots from 50k training samples with 50 training repetitions.

The 3D CNN architecture of MoviNet exhibited a clear scaling effect under the limited-data regime of 50k training samples. At the 3k and 5k parameter scales, MoviNet slightly underperformed against the benchmark, with median F1 differences ranging from approxi-

mately −1% to −0.5%, likely reflecting the high degrees of freedom associated with joint 3D spatiotemporal convolutions, requiring sufficient capacity to be effectively utilized. At the 10k scale, MoviNet achieved a marked relative improvement, surpassing the benchmark with a median F1 difference of 6%. This gain is partially attributable to a degradation in benchmark performance at higher parameter counts under limited-data conditions, while MoviNet remains comparatively stable. These results suggest that under limited-data conditions, MoviNet exhibits robustness to increasing parameter budgets and is better able to leverage additional capacity to model holistic spatiotemporal features when benchmark architectures struggle to do so. Conversely, TVN exhibited a non-monotonic performance trend under the limited-data condition. The median F1 difference improved slightly from approximately −0.8% to −0.2% from 3k to 5k, followed by a decline to approximately −8% at the 7.5k scale. At the 10k scale, TVN showed a partial relative recovery; however, this improvement coincides with a degradation in benchmark performance. This pattern suggests that TVN is sensitive to parameter sizes under limited-data conditions, where increases in model size do not consistently translate into improved generalization. While additional parameters may alleviate underfitting at very low capacities, they may also exacerbate optimization instability or overfitting when training data are insufficient.

5. Discussion and Limitations

Across the 25 training repetitions and five model sizes, the SRU-based lightweight hybrids (MobileNetSRU and ConvNextKanSRU) were the best performers overall, consistently beating the CNNTransformer benchmark at the small-to-medium parameter scales and retaining competitive performance at larger model sizes as shown in Figure 3. Several interacting factors likely explain why SRU-based hybrids fit our data better than other models. First, the inductive bias (i.e., the assumption that an algorithm uses to generalize from training data to unseen data) of RNNs, assuming past observations are relevant to current predictions, aligned with the annual phenology of the Landsat sequences. The Landsat time series in this study exhibit three key characteristics: first, moderate sequence length (yearly sequences); second, strong but regular temporal dynamics driven by phenology; and third, class-dependent spectral stability (e.g., water vs. wetland). Under these conditions, SRU-based models benefit from the inherent sequential inductive bias that directly encodes temporal order and continuity, allowing efficient modeling of phenological trajectories with limited parameters [46,67]. In contrast, transformers with self-attention mechanisms, as well as the lightweight linear-attention variants, must learn temporal relations more flexibly. Unlike RNNs, they do not assume data is sequential by default; instead, they process all time steps simultaneously to identify global dependencies [68]. This flexibility allows them to start with fewer assumptions, making them more general but also requiring more capacity or larger model sizes and more data to discover and capture the sequential patterns that the SRU already has built-in [51]. Second, with the parameter-efficient design of compact set of gates and parallelizable linear transforms, the SRU achieves good temporal modeling with few parameters, reducing overfitting on classes with limited or noisy signals and benefiting small models [69]. Third, SRUs benefit from robustness to temporal irregularity and missing observations [46]. We applied cloud filtering to the Landsat sequences, and these sequences also contain sensor transitions (L5, L7, L8/9). The SRU's sequential gating and localized temporal dynamics tolerate uneven or sparse observations more naturally than attention mechanisms, which assume dense (i.e., observations are frequent with very few or no missing observations over the sequence), and stable (i.e., time intervals between observations are fixed and regular) temporal relationships [70]. The observed advantage of the SRU-based architecture in this study is consistent with prior findings that recurrent models are well-suited for satellite

image time-series classification. For instance, research has demonstrated that recurrent designs effectively capture seasonal dynamics in multi-temporal data [71] and outperform traditional classifiers in complex tasks such as crop mapping [72]. While transformer-based models can achieve superior performance when provided with sufficient capacity and massive datasets [9,73], they often require larger parameter scales to fully exploit their attention mechanisms [74,75].

A notable pattern observed in Figure 4 is that as model size increased, the CNNTransformer benchmark progressively closed the performance gap with the SRU-based models. This suggests that RNN-based temporal modules are less able to capitalize on increased model capacity, whereas transformer layers scale more effectively. This observation is consistent with prior findings in sequence modeling, where recurrent architectures tend to saturate in performance as depth increases, while attention-based models benefit more from scaling in both width and depth [9,76]. RNNs must process sequences sequentially, limiting parallelism, increasing gradient decay over longer dependencies, and reducing their ability on deeper architectures. These limitations have been widely discussed in the literature on sequence learning, particularly in the context of long-term dependency modeling [77,78]. In contrast, transformer architectures are explicitly designed to scale: their parallel attention mechanism enables more effective use of additional parameters, supports the modeling of long-range temporal relationships, and maintains stable gradients as depth increases [9]. These scaling advantages are widely recognized in other domains, most notably in NLP, where transformers have replaced RNNs as the foundation of modern LLMs [60]. The lightweight video transformer (EVT) struggled to form stable and meaningful attention patterns at small model sizes, leading to low performance and high variability as shown in Figure 3. This behavior is consistent with prior studies indicating that transformer-based models are sensitive to data availability and model size [74]. However, as model size increased, the EVT showed clear improvements, suggesting that attention-based architectures may require greater capacity to effectively model the temporal structure of Landsat time series. With larger datasets or substantial pretraining, EVT may eventually match or surpass the CNNTransformer benchmark. Such scaling behavior has also been reported in both the computer vision and machine learning literature, where transformer performance improves significantly with increased training data and model complexity [79]. A similar pattern is observed in the 3D CNN family: MoviNet outperformed the 2D + 1D TVN at larger model sizes. One possible reason is that MoviNet's entirely joint spatiotemporal convolutions can better exploit additional parameters to learn richer and more holistic spatial–temporal features. This aligns with prior work showing that joint spatiotemporal feature learning can outperform factorized approaches when sufficient model capacity is available [80,81]. By analogy, a higher-capacity EVT, capable of jointly capturing spatial and temporal relationships within its attention mechanism, may ultimately outperform architectures that separate spatial encoding (CNN) and temporal modeling (transformer), provided there it has a sufficient model size and available training data. Besides jointly capturing spatial and temporal relationships, the video transformer architecture can perform dynamic token weighting across both space and time [82], while the CNN + transformer architecture cannot perform this dynamic cross-dimensional weighting until after fixed spatial features are extracted by the CNN. This flexibility has been identified as a key advantage of attention mechanisms in modeling complex dependencies across dimensions [9]. Moreover, while CNN spatial encoders excel at modeling local spatial patterns, attention-based mechanisms are superior at capturing global relationships [9]. This distinction between local inductive bias (CNN) and global context modeling (attention) has been widely documented in both the computer vision and remote sensing literature [74,83]. This capacity for global context

is crucial in large-scale RS LCLU classification, as local convolutional operations often struggle to integrate distant spatial information without extensive network depth [84].

The transition from a 500k to a 50k training sample size suggests the critical role of inductive bias in maintaining model robustness when empirical evidence is sparse. While the CNNTransformer benchmark exhibited performance degradation, MobileNetSRU demonstrated resilience. This divergence reflects the sample inefficiency that is characteristic of transformer architectures, which typically rely on large training datasets to learn meaningful attention weights and stable temporal representations. In the absence of sufficient data, attention mechanisms struggle to reliably identify important time steps and long-range dependencies, particularly at small parameter scales where representational capacity is constrained [9,85,86]. Conversely, the sequential inductive biases of the SRU provide a regularizing framework, enabling the model to maintain temporal representation despite limited observations [87].

Another important mechanism underlying the superior performance of the SRU-based lightweight models is the separation between spatial encoding and temporal aggregation. In the convolutional and recurrent hybrid architectures, the convolutional backbone first compresses the 7×7 Landsat patches into compact spatial representations before temporal modeling is applied. This staged design reduces the dimensionality of the temporal learning problem and allows the recurrent module to focus primarily on phenological evolution rather than simultaneously resolving spatial texture and temporal relationships. In contrast, architectures such as 3D CNNs and video transformers attempt to jointly model spatial and temporal interactions from the beginning of the network [12,60], substantially increasing optimization complexity at small parameter scales. Under constrained parameter budgets, the SRU-based hybrids therefore allocate model capacity more efficiently by decomposing the learning process into two simpler subtasks of spatial feature extraction and temporal sequence modeling.

The SRU mechanism itself also provides several advantages for medium-spatial-resolution optical satellite image time series. Unlike traditional recurrent architectures such as LSTM or GRU, the SRU removes most recurrent dependencies from the heavy matrix multiplication operations, enabling parallel computation across temporal steps while still preserving sequential gating behavior [46]. This design improves computational efficiency and stabilizes optimization, particularly when yearly Landsat sequences contain irregular temporal gaps caused by cloud masking, sensor transitions, or variable acquisition frequency. Because the hidden-state interaction in the SRU is relatively lightweight, the architecture can preserve temporal continuity without requiring the large parameter budgets typically needed for transformer-based architectures. This property is especially important in Landsat applications where the temporal signal is often smoother and more seasonally structured. Consequently, the SRU mechanism appears well aligned with the moderate temporal complexity and relatively stable annual phenology of Landsat observations.

The behavior of the transformer-based lightweight architectures further highlights the importance of matching architectural assumptions to the statistical structure of the data. The lightweight transformer hybrids showed underperformance at the smallest parameter scales, despite transformers being highly successful in other domains [55,62]. One explanation is that self-attention mechanisms rely on learning flexible pairwise temporal relationships directly from data [9], but the limited parameter budgets at small scales may be insufficient to construct stable and discriminative attention patterns. Additionally, linear-attention approximations improve computational efficiency by compressing temporal interactions [59], but this compression may discard the subtle phenological relationships needed for separating spectrally similar and temporally heterogeneous classes. As model size increased, the transformer-based models gradually improved, supporting

the interpretation that attention mechanisms become increasingly effective once sufficient representational capacity is available.

Despite the strong performance of the SRU-based lightweight architectures, several model-specific limitations should also be acknowledged. First, recurrent architectures inherently process temporal information sequentially, which may limit their ability to capture very-long-range dependencies compared to transformer-based architectures with global self-attention mechanisms. While the SRU improves computational parallelization relative to traditional recurrent models, its temporal modeling capacity may still saturate as model complexity increases, as observed in the diminishing relative gains at larger parameter scales. Second, the lightweight CNN backbone prioritizes parameter efficiency and local spatial feature extraction, which may reduce the ability to capture broader spatial context and complex landscape structures compared to heavier convolutional or fully attention-based architectures.

In addition, lightweight architectures involve an inherent trade-off between efficiency and representational flexibility. Mechanisms designed to reduce computational complexity, such as depthwise convolutions, compact gating structures, and linear-attention approximations, can improve parameter efficiency but may also restrict the richness of learned spatial–temporal representations. Consequently, lightweight models may become less effective when classification tasks require modeling highly irregular temporal dynamics, subtle class boundaries, or complex multi-scale spatial interactions. Although the present study demonstrates that lightweight architectures can achieve strong performance for yearly Landsat time-series classification, the extent to which these models generalize to denser temporal observations, higher spatial resolutions, or more heterogeneous ecological systems remains uncertain.

Our findings should be interpreted within the context of several limitations. First, the training samples used in this study were fixed to 7×7-pixel neighborhoods. Altering this size, either by expanding the patch to incorporate a wider spatial context or by reducing it to 3×3 or 5×5 to assess finer local feature extraction, could potentially alter the spatial discrimination capabilities of the tested architectures. Second, the temporal inputs in this study were restricted to single-year Landsat sequences. While this captures intra-annual phenology, it does not evaluate the models' ability to handle inter-annual variation, disturbance and recovery trajectories. Third, although the dataset spans diverse CONUS ecoregions, generalization to other global ecological zones (e.g., tropical or arid systems) remain untested, where the phenological patterns and spectral separability may differ. Fourth, this study focuses only on Landsat data; integrating multi-source data (e.g., Sentinel-2, SAR, DEM, or climate variables) may change model rankings by enhancing spatial resolution, temporal density, or feature richness. Future work should systematically evaluate cross-region transferability and multi-source fusion to better understand the robustness of lightweight architectures under more heterogeneous environmental conditions.

Another limitation is the use of parameter count as the primary measure of model efficiency. While parameter size provides a useful and architecture-independent measure for fair comparison, it does not fully capture operational efficiency in practical deployment scenarios. Different lightweight architectures may exhibit substantially different inference latency, memory access patterns, or hardware utilization characteristics even when parameter counts are similar. For example, recurrent operations, depthwise convolutions, and attention mechanisms may behave differently on GPUs, CPUs, or edge devices due to differences in parallelization efficiency and memory bandwidth requirements. Consequently, models with comparable parameter counts may still differ substantially in real-world inference speed and energy consumption. Future studies should therefore include hardware-

aware benchmarking metrics such as inference latency, floating point operations (FLOPs), memory footprint, and power consumption across multiple deployment environments.

The current study evaluates yearly Landsat sequences as independent samples and does not explicitly model long-term ecological trajectories or abrupt disturbance events. Processes such as wildfire recovery, urban expansion, forest harvesting, or long-term wetland transitions often evolve across multiple years or decades. While yearly sequences capture intra-annual phenology effectively, they may not fully represent slower ecological dynamics that require multi-year temporal context. Similarly, the study focuses exclusively on classification performance and does not investigate model interpretability or uncertainty estimation. Understanding which temporal observations or spatial regions contribute most strongly to lightweight model predictions could provide valuable ecological insight and improve reliability for operational Earth-observation applications. Future work should investigate interpretable lightweight architectures, uncertainty-aware prediction frameworks, and long-term temporal modeling strategies for large-scale remote sensing monitoring tasks.

6. Conclusions

This study demonstrates that SRU-based lightweight hybrids, specifically MobileNetSRU and ConvNextKanSRU, consistently outperform the CNNTransformer benchmark for Landsat LCLU classification at small-to-medium scales (3k to 15k parameters). The success of these models is attributed to a strong inductive bias that aligns with the sequential phenology of yearly Landsat data, providing superior robustness and stability, particularly when training data is scarce. MobileNetSRU achieves peak relative improvements of 2.5–7.5% at the 7.5k parameter scale and excels in spectrally stable classes like water and bare land, but its advantage diminishes as complexity increases. At larger scales (25k–50k parameters), the CNNTransformer scales more effectively for complex vegetation dynamics such as forests and wetlands, though ConvNextKanSRU remains competitive.

Ultimately, the choice of a lightweight model must balance architectural stability with the specific requirements of the LCLU task. While models like MobileNetSRU offer an efficient and high-performance solution for resource-constrained or data-limited environments, such as cloud-based continental monitoring or edge-processing scenarios, larger hybrid architectures remain preferable for monitoring complex ecological transitions when computational resources are abundant. Future research should explore the impact of varying spatial patch sizes and expand these lightweight frameworks to longer time series and more geographically diverse regions. Such advancements will be critical in determining if these efficiency gains hold under broader environmental variability and increased model complexity for large-scale remote sensing applications.

Author Contributions: Conceptualization, Z.W. and G.M.; methodology, G.M. and Z.W.; data, Z.W. and A.S.; software, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, G.M. and Z.W.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study will be available on request.

Acknowledgments: We thank the OrangeGrid high-throughput computing (HTC) cluster at Syracuse University for providing essential computational resources.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rahman, A.; Kumar, S.; Fazal, S.; Siddiqui, M.A. Assessment of land use/land cover change in the North-West District of Delhi using remote sensing and GIS techniques. *J. Indian Soc. Remote Sens.* **2012**, *40*, 689–697. [CrossRef]
2. Talukdar, S.; Singha, P.; Mahato, S.; Pal, S.; Liou, Y.A.; Rahman, A. Land-use land-cover classification by machine learning classifiers for satellite observations—A review. *Remote Sens.* **2020**, *12*, 1135. [CrossRef]
3. Pande, C.B.; Srivastava, A.; Moharir, K.N.; Radwan, N.; Mohd Sidek, L.; Alshehri, F.; Pal, S.C.; Tolche, A.D.; Zhran, M. Characterizing land use/land cover change dynamics by an enhanced random forest machine learning model: A Google Earth Engine implementation. *Environ. Sci. Eur.* **2024**, *36*, 84. [CrossRef]
4. Gashaw, T.; Tulu, T.; Argaw, M.; Worqlul, A.W. Evaluation and prediction of land use/land cover changes in the Andassa watershed, Blue Nile Basin, Ethiopia. *Environ. Syst. Res.* **2017**, *6*, 17. [CrossRef]
5. Pettorelli, N. *Satellite Remote Sensing and the Management of Natural Resources*; Oxford University Press: Oxford, UK, 2019.
6. De Leeuw, J.; Georgiadou, Y.; Kerle, N.; De Gier, A.; Inoue, Y.; Ferwerda, J.; Smies, M.; Narantuya, D. The function of remote sensing in support of environmental policy. *Remote Sens.* **2010**, *2*, 1731–1750. [CrossRef]
7. Giezen, M.; Balikci, S.; Arundel, R. Using remote sensing to analyse net land-use change from conflicting sustainability policies: The case of Amsterdam. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 381. [CrossRef]
8. Persello, C.; Wegner, J.D.; Hänsch, R.; Tuia, D.; Ghamisi, P.; Koeva, M.; Camps-Valls, G. Deep learning and earth observation to support the sustainable development goals: Current approaches, open challenges, and future opportunities. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 172–200. [CrossRef]
9. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
10. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15908–15919.
11. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244. [CrossRef]
12. Wang, J.; Yang, X.; Li, H.; Liu, L.; Wu, Z.; Jiang, Y.G. Efficient video transformers with spatial-temporal token selection. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 69–86.
13. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
14. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
15. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
16. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
17. Tong, W.; Chen, W.; Han, W.; Li, X.; Wang, L. Channel-attention-based DenseNet network for remote sensing image scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 4121–4132. [CrossRef]
18. Liu, B.; Bai, Y. Improving Scarce RS Data Classification with Independent Noise and Feature Mutual Exclusion. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 6009205. [CrossRef]
19. Song, H.; Wei, C.; Yong, Z. Efficient knowledge distillation for remote sensing image classification: A CNN-based approach. *Int. J. Web Inf. Syst.* **2023**, *19*, 153–167. [CrossRef]
20. Baumgardner, M.; Biehl, L.; Landgrebe, D. 220 band AVIRIS hyperspectral image data set: June 12, 1992, Indian pine test site 3. In *Purdue University Research Repository*; Purdue University: West Lafayette, IN, USA, 2015. [CrossRef]
21. Xu, X.; Li, J.; Plaza, A. Fusion of hyperspectral and LiDAR data using morphological component analysis. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*; IEEE: New York, NY, USA, 2016; pp. 3575–3578.
22. Qin, Y.; Bruzzone, L.; Li, B. Tensor alignment based domain adaptation for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9290–9307. [CrossRef]
23. Pande, S.; Banerjee, B. Adaptive hybrid attention network for hyperspectral image classification. *Pattern Recognit. Lett.* **2021**, *144*, 6–12. [CrossRef]
24. Das, A.; Saha, I.; Scherer, R. GhoMR: Multi-receptive lightweight residual modules for hyperspectral classification. *Sensors* **2020**, *20*, 6823. [CrossRef]
25. Hu, X.; Yang, W.; Wen, H.; Liu, Y.; Peng, Y. A lightweight 1-D convolution augmented transformer with metric learning for hyperspectral image classification. *Sensors* **2021**, *21*, 1751. [CrossRef]
26. Arshad, T.; Zhang, J. A light-weighted spectral-spatial transformer model for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 4567–4579. [CrossRef]

27. Wang, J.; Hu, J.; Liu, Y.; Hua, Z.; Hao, S.; Yao, Y. Elnas: Efficient lightweight attention cross-domain architecture search for hyperspectral image classification. *Remote Sens.* **2023**, *15*, 4688. [CrossRef]
28. Wang, Y.; Li, S.; Lin, Y.; Wang, M. Lightweight deep neural network method for water body extraction from high-resolution remote sensing images with multisensors. *Sensors* **2021**, *21*, 7397. [CrossRef]
29. Liu, S.; Cao, S.; Lu, X.; Peng, J.; Ping, L.; Fan, X.; Teng, F.; Liu, X. Lightweight Deep Learning Model, ConvNeXt-U: An Improved U-Net Network for Extracting Cropland in Complex Landscapes from Gaofen-2 Images. *Sensors* **2025**, *25*, 261. [CrossRef]
30. Zhang, M.; An, J.; Yang, L.D.; Wu, L.; Lu, X.Q. Convolutional neural network with attention mechanism for SAR automatic target recognition. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 4004205.
31. Geng, X.; Zhao, L.; Shi, L.; Yang, J.; Li, P.; Sun, W. Small-sized ship detection nearshore based on lightweight active learning model with a small number of labeled data for sar imagery. *Remote Sens.* **2021**, *13*, 3400. [CrossRef]
32. Yu, H.; Wang, C.; Li, J.; Sui, Y. Automatic extraction of green tide from GF-3 SAR images based on feature selection and deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 10598–10613. [CrossRef]
33. Mazzia, V.; Khaliq, A.; Chiaberge, M. Improvement in land cover and crop classification based on temporal features learning from Sentinel-2 data using recurrent-convolutional neural network (R-CNN). *Appl. Sci.* **2019**, *10*, 238. [CrossRef]
34. Garnot, V.S.F.; Landrieu, L. Lightweight temporal self-attention for classifying satellite images time series. In *International Workshop on Advanced Analytics and Learning on Temporal Data*; Springer: Cham, Switzerland, 2020; pp. 171–181.
35. Corbane, C.; Syrris, V.; Sabo, F.; Politis, P.; Melchiorri, M.; Pesaresi, M.; Soille, P.; Kemper, T. Convolutional neural networks for global human settlements mapping from Sentinel-2 satellite imagery. *Neural Comput. Appl.* **2021**, *33*, 6697–6720. [CrossRef]
36. Arrechea-Castillo, D.A.; Solano-Correa, Y.T.; Muñoz-Ordóñez, J.F.; Pencue-Fierro, E.L.; Figueroa-Casas, A. Multiclass land use and land cover classification of Andean Sub-Basins in Colombia with Sentinel-2 and Deep Learning. *Remote Sens.* **2023**, *15*, 2521. [CrossRef]
37. Papoutsis, I.; Bountos, N.I.; Zavras, A.; Michail, D.; Tryfonopoulos, C. Benchmarking and scaling of deep learning models for land cover image classification. *ISPRS J. Photogramm. Remote Sens.* **2023**, *195*, 250–268. [CrossRef]
38. Sawant, S.; Ghosh, J.K. Land use land cover classification using Sentinel imagery based on deep learning models. *J. Earth Syst. Sci.* **2024**, *133*, 101. [CrossRef]
39. Wang, Y.; Feng, L.; Sun, W.; Wang, L.; Yang, G.; Chen, B. A lightweight CNN-Transformer network for pixel-based crop mapping using time-series Sentinel-2 imagery. *Comput. Electron. Agric.* **2024**, *226*, 109370. [CrossRef]
40. Sencaki, D.B.; Putri, M.N.; Santosa, B.H.; Arfah, S.; Arifandri, R.; Afifuddin; Habibie, M.I.; Putra, P.K.; Anatoly, N.; Permata, Z.D.O.; et al. Land cover multiclass classification of Wonosobo, Indonesia with time series-based one-dimensional deep learning model. *Remote Sens. Appl. Soc. Environ.* **2023**, *32*, 101040. [CrossRef]
41. Wan, J.; Yong, B. Automatic extraction of surface water based on lightweight convolutional neural network. *Ecotoxicol. Environ. Saf.* **2023**, *256*, 114843. [CrossRef]
42. Martono, D.N.; Santosa, B.H.; Sencaki, D.B.; Arifandri, R.; Hakim, A.M.Y.; Prayogi, H.; Apip; Gandharum, L.; Steinhausen, M.J.; Schröter, K. Enhanced light 1D-based Deep Learning algorithm for land cover classification in Citarum upper watershed, Indonesia. *Remote Sens. Appl. Soc. Environ.* **2025**, *37*, 101773. [CrossRef]
43. Stehman, S.V.; Pengra, B.W.; Horton, J.A.; Wellington, D.F. Validation of the US Geological Survey's Land Change Monitoring, Assessment and Projection (LCMAP) Collection 1.0 annual land cover products 1985–2017. *Remote Sens. Environ.* **2021**, *265*, 112646. [CrossRef]
44. Pengra, B.W.; Stehman, S.V.; Horton, J.A.; Dockter, D.J.; Schroeder, T.A.; Yang, Z.; Hernandez, A.J.; Healey, S.P.; Cohen, W.B.; Finco, M.V.; et al. *LCMAP Reference Data Product 1984–2018 Land Cover, Land Use and Change Process Attributes (ver. 1.2)*; U.S. Geological Survey: Reston, VA, USA, 2020. [CrossRef]
45. Mountrakis, G.; Heydari, S.S. Harvesting the Landsat archive for land cover land use classification using deep neural networks: Comparison with traditional classifiers and multi-sensor benefits. *ISPRS J. Photogramm. Remote Sens.* **2023**, *200*, 106–119. [CrossRef]
46. Lei, T.; Zhang, Y.; Wang, S.I.; Dai, H.; Artzi, Y. Simple recurrent units for highly parallelizable recurrence. *arXiv* **2018**, arXiv:1709.02755. [CrossRef]
47. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
48. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
49. Ghazal, M.; Waisi, N.; Abdullah, N. The detection of handguns from live-video in real-time based on deep learning. *TELKOMNIKA* **2020**, *18*, 3026–3032. [CrossRef]
50. Zhou, G.; Liu, W.; Zhu, Q.; Lu, Y.; Liu, Y. ECA-MobileNetV3 (Large)+ SegNet model for binary sugarcane classification of remotely sensed images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4414915. [CrossRef]

51. Pan, J.; Lei, T.; Kim, K.; Han, K.J.; Watanabe, S. SRU++: Pioneering fast recurrence with attention for speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2022; pp. 7872–7876.
52. Cheon, M. Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing. *arXiv* **2024**, arXiv:2406.00600. [CrossRef]
53. Woo, S.; Debnath, S.; Hu, R.; Chen, X.; Liu, Z.; Kweon, I.S.; Xie, S. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 16133–16142.
54. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.; Tegmark, M. Kan: Kolmogorov-arnold networks. *arXiv* **2024**, arXiv:2404.19756.
55. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef]
56. Wang, R.; Jiang, H.; Li, Y. Upernet with convnext for semantic segmentation. In *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*; IEEE: New York, NY, USA, 2023; pp. 764–769.
57. Zhou, J.; Zhang, B.; Yuan, X.; Lian, C.; Ji, L.; Zhang, Q.; Yue, J. YOLO-CIR: The network based on YOLO and ConvNeXt for infrared object detection. *Infrared Phys. Technol.* **2023**, *131*, 104703. [CrossRef]
58. Jamali, A.; Roy, S.K.; Hong, D.; Lu, B.; Ghamisi, P. How to learn more? Exploring Kolmogorov–Arnold networks for hyperspectral image classification. *Remote Sens.* **2024**, *16*, 4015. [CrossRef]
59. Arora, S.; Eyuboglu, S.; Zhang, M.; Timalsina, A.; Alberti, S.; Zinsley, D.; Zou, J.; Rudra, A.; Ré, C. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv* **2024**, arXiv:2402.18668.
60. Kondratyuk, D.; Yuan, L.; Li, Y.; Zhang, L.; Tan, M.; Brown, M.; Gong, B. Movinets: Mobile video networks for efficient video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16020–16030.
61. Piergiovanni, A.J.; Angelova, A.; Ryoo, M.S. Tiny video networks. *Appl. AI Lett.* **2022**, *3*, e38. [CrossRef]
62. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A survey of large language models. *arXiv* **2023**, arXiv:2303.18223. [CrossRef]
63. Zela, A.; Klein, A.; Falkner, S.; Hutter, F. Towards automated deep learning: Efficient joint neural architecture and hyperparameter search. *arXiv* **2018**, arXiv:1807.06906. [CrossRef]
64. Melyani, N.A.; Lubis, A.F.; Tatamara, A.; Haiban, R.R.; Iltizam, M.; Rofiqi, M.A.; Abdurrahman, S.H.; Samae, N.; Shahid, B.; Habibullah, M.; et al. Analysis Comparison Classification Image Disease Eye Using the CNN Algorithm, Inception V3, DenseNet 121 and MobileNet V2 Architecture Models. *Public Res. J. Eng. Data Technol. Comput. Sci.* **2025**, *3*, 42–58. [CrossRef]
65. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; Schmid, C. Vivit: A video vision transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 6836–6846.
66. Liang, Y.; Zhou, P.; Zimmermann, R.; Yan, S. DualFormer: Local-Global Stratified Transformer for Efficient Video Recognition. *arXiv* **2021**, arXiv:2112.04674.
67. Medsker, L.R.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 64–67.
68. Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A survey of transformers. *AI Open* **2022**, *3*, 111–132. [CrossRef]
69. Chen, Y.; Li, J.; Guo, N. Efficient and interpretable SRU combined with TabNet for network intrusion detection in the big data environment. *Int. J. Inf. Secur.* **2023**, *22*, 679–689. [CrossRef]
70. Du, W.; Côté, D.; Liu, Y. Saits: Self-attention-based imputation for time series. *Expert Syst. Appl.* **2023**, *219*, 119619. [CrossRef]
71. Rußwurm, M.; Körner, M. Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 129. [CrossRef]
72. Pelletier, C.; Webb, G.I.; Petitjean, F. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sens.* **2019**, *11*, 523. [CrossRef]
73. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
74. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
75. d'Ascoli, S.; Touvron, H.; Leavitt, M.L.; Morcos, A.S.; Biroli, G.; Sagun, L. Convit: Improving vision transformers with soft convolutional inductive biases. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 2286–2296.
76. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.

77. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
78. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
79. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling laws for neural language models. *arXiv* **2020**, arXiv:2001.08361. [CrossRef]
80. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6202–6211.
81. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
82. Wang, C.H.; Huang, K.Y.; Yao, Y.; Chen, J.C.; Shuai, H.H.; Cheng, W.H. Lightweight deep learning: An overview. *IEEE Consum. Electron. Mag.* **2022**, *13*, 51–64. [CrossRef]
83. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
84. Yang, J.; Li, C.; Zhang, P.; Dai, X.; Xiao, B.; Yuan, L.; Gao, J. Focal attention for long-range interactions in vision transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 30008–30022.
85. Sharir, G.; Noy, A.; Zelnik-Manor, L. An image is worth 16x16 words, what is a video worth? *arXiv* **2021**, arXiv:2103.13915. [CrossRef]
86. Zhai, X.; Kolesnikov, A.; Houlsby, N.; Beyer, L. Scaling vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12104–12113.
87. Weerakody, P.B.; Wong, K.W.; Wang, G.; Ela, W. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing* **2021**, *441*, 161–178. [CrossRef]