



Lightweight and Efficient Deep Learning Models for Remote Sensing-Based Land Use and Land Cover Classification: A Case Study on EuroSAT Dataset

Mohammad Alikhani¹ , Behnam Ebadati² , and Reza Attarzadeh³

1. Faculty of Geodesy and Geomatics Engineering, K. N. Toosi University of Technology, Tehran, Iran. E-mail: m.alikhani1@email.kntu.ac.ir
2. Corresponding author, Department of Geomatics Engineering, ST.C., Islamic Azad University, Tehran, Iran. E-mail: st_b_ebadati@azad.ac.ir
3. Department of Geomatics Engineering, ST.C., Islamic Azad University, Tehran, Iran. E-mail: reza.attarzadeh@iau.ac.ir

Article Info

Article type:
Research Article

Article history:
Received 2025-05-30
Received in revised form 2025-07-21
Accepted 2025-07-22
Published online 2025-10-19

Keywords:
Remote Sensing,
Sentinel-2,
Multispectral,
LULC Classification,
Deep Learning

ABSTRACT

One of the primary applications of remote sensing is the classification of land use and land cover (LULC). This field has increased prominence in computer vision tasks with the expansion of learning methods, especially in recent years. In this context, the present study introduces a convolutional neural network architecture named HCNN, designed to achieve high accuracy in LULC classification while minimizing processing cost. A simpler architecture based on conventional CNN designs, referred to as SCNN, was also implemented for comparison. The HCNN architecture comprises residual, dense, inception, and squeeze-excitation blocks, along with several base layers. Four models based on these two architectures were trained using RGB and multispectral data from the Sentinel-2 imagery in the EuroSAT dataset: SCNN-RGB, SCNN-MS, HCNN-RGB, and HCNN-MS. All models achieved accuracies above 94%. Among these, HCNN-MS attained the highest accuracy of 98.44%, while SCNN-RGB recorded the lowest accuracy at 95.57%. Overall, HCNN-based models outperformed SCNN-based models in accuracy and training speed, requiring approximately six times less training time. Additionally, the use of multispectral data had a positive impact on model accuracy, albeit it increased computational complexity somewhat. An ablation study was also conducted to evaluate the role of each block in the HCNN architecture on the model's final performance. The ablation results demonstrated that each block plays a significant role in improving accuracy and reducing processing overhead, particularly the residual and dense blocks, which had the greatest impact on final accuracy. Moreover, the squeeze-excitation block notably reduced training time, while its removal caused minimal change in accuracy.

Cite this article: Last Name, Initial., Last Name, Initial., & Last Name, Initial. (2025). Lightweight and Efficient Deep Learning Models for Remote Sensing-Based Land Use and Land Cover Classification: A Case Study on EuroSAT Dataset, Earth Observation and Geomatics Engineering, Volume 8, Issue 2, Pages 106-118. <http://doi.org/10.22059/eoge.2025.396331.1179>



© The Author(s).

Publisher: University of Tehran.

DOI: <http://doi.org/10.22059/eoge.2025.396331.1179>

1. Introduction

Remote sensing plays a crucial role in Earth observation, enabling the continuous monitoring of environmental changes, land use patterns, and natural resource management (Panda et al., 2024; Sharma et al., 2024; Li et al., 2021). It provides essential data for applications such as climate change assessment, urban planning, disaster response, and agricultural monitoring (Filchev & Chanev, 2024; Mohan et al., 2024; Padmaja et al., 2024). Remote sensing utilizes satellite and airborne sensors to capture electromagnetic radiation reflected or emitted from the Earth's surface. Therefore, this provides a detailed perspective on terrestrial and atmospheric conditions (Ghimire). Advances in remote sensing technology have greatly improved the spatial, spectral, and temporal resolution of satellite imagery, enabling more precise analyses (Bing, 2017; Lian et al., 2025; Jiang et al., 2021; Swain et al., 2024). Furthermore, the increasing availability of high-resolution multispectral data has enabled researchers to extract valuable information from large datasets (Lim et al., 2024).

However, accurate classification tasks such as land cover mapping using images derived from satellite sensors are a key challenge in remote sensing (Fayaz et al., 2024). Therefore, extracting meaningful features from these images is crucial, especially for multispectral data. The multispectral data provides valuable spectral and spatial information for achieving accurate results. Traditional classification approaches have relied on manually engineered features and shallow learning methods. These include spectral, texture, and object-based analyses (Blaschke, 2013; Zeng et al., 2021). However, these methods are limited by their dependence on domain expertise and manual work. They are also vulnerable to lighting, scale, and atmospheric changes (Zhu et al., 2017). Additionally, differences in sensor types, spatial resolutions, and spectral bands pose challenges in creating robust classification techniques (Chen et al., 2018; Li et al., 2023).

The increasing complexity and volume of remote sensing data require advanced methods for processing high-dimensional imagery. Neural networks offer a promising solution to handle these challenges (Alyahyan, 2025) efficiently. Convolutional Neural Networks (CNNs) are widely used in computer vision tasks. Their extension to remote sensing has shown remarkable success in analyzing high-dimensional satellite data (Chen et al., 2020; Liu et al., 2020). The CNNs can identify spatial patterns and extract spectral relationships between pixels, especially in multispectral remote sensing data. Recent studies have demonstrated their effectiveness in remote sensing tasks, including land cover classification, object detection, and scene recognition (Lyons et al., 2018; Zeng et al., 2018; Shi et al., 2020; Tiwari & Shukla, 2018; Guo et al., 2021).

Despite the advantages of CNNs, their use in remote sensing faces certain challenges. Training neural network models, especially in deep learning, requires a large volume of labeled data, which presents a significant challenge (Li et al., 2023; Yuan et al., 2023). However, with efforts from researchers, valuable benchmark datasets have been made publicly available, partially addressing this challenge. For instance, many studies have used datasets such as EuroSAT for land use and land cover classification (Bhatti et al., 2025; Mohammed & Lakizadeh, 2025; Niu et al., 2025; Stival et al., 2025). Another challenge is the predominant focus on three-channel models in computer vision, which are designed primarily for RGB data. As a result, the rich spectral features provided by multispectral remote sensing data have been largely overlooked (Yassine et al., 2021). Well-known models, such as VGG-16 and ResNet-50, which have been trained on the ImageNet database, rely on three-channel inputs in their initial architecture (Jannat and Hossain, 2024). Optimizing the computational efficiency of CNNs while maintaining or improving accuracy, especially when handling high-dimensional multispectral data, remains a key challenge.

Most studies have focused on enhancing CNN accuracy and overall performance. However, metrics such as feature extraction speed and training time, which are crucial for computational efficiency, have received less attention. For instance, in a study conducted by Acuña-Alonso, a CNN model was trained on the EuroSAT dataset for LULC classification. This model achieved an accuracy of 88% on the test data, demonstrating satisfactory classification performance (Acuña-Alonso et al., 2024). In another study, Yamashkin developed the GeosystemNet for situations with limited data. In this study, when only 10% of the data was used for training and 90% for testing, the model achieved an accuracy of 89.23% (Yamashkin et al., 2020). In another study, Yassine trained a CNN model on the multispectral EuroSAT dataset instead of its RGB version and reported an overall accuracy of 98.78%. Additionally, computing spectral indices and incorporating them into the input data further improved the accuracy to 99.58% (Yassine et al., 2021). These studies confirm CNNs' performance in remote sensing tasks while not reporting on their computational efficiency.

On the other hand, some studies have focused on improving the computational efficiency of CNNs and have provided reports in this regard. For example, Albarakati proposed a method combining a self-attention mechanism with integrating IBNR-65 and DenseNet-64 networks. The results indicated that the processing time was nearly twice as fast as VGG-16 while maintaining an accuracy of 89.5% (Albarakati et al., 2024). In another study, researchers employed dimension reduction techniques, such as histogram sampling and the Bag of Words method, to improve the efficiency of CNNs trained on the BigEarthNet-

S2 dataset while preserving their classification accuracy and performance (Calota et al., 2015). It can be inferred that researchers primarily focus on developing highly accurate CNNs rather than optimizing for highly accurate results and computational efficiency.

Based on the aforementioned points, the main objective of this study is to design an architecture capable of flexibly utilizing both RGB and multispectral data with a customizable number of channels. In addition to achieving significant accuracy, this architecture should also optimize computational efficiency and strike an effective balance between accuracy and performance. The proposed architecture was applied to LULC classification at the patch level, where each image patch is labeled based on its dominant LULC class. Unlike pixel- or object-level classification, this approach is not designed for producing thematic maps or for annotating specific objects such as buildings or trees. The evaluation of results will provide insight into how the CNN architecture and input data influence LULC classification performance, with a focus on achieving high accuracy while maintaining reduced processing demand.

2. Methodology

2.1. Data Description

The EuroSAT dataset is widely used as a benchmark for land use and land cover (LULC) classification/detection tasks. Numerous studies have utilized this dataset for developing deep learning and machine learning models (Günen et al., 2022; Kunwar et al., 2023; Loganathan et al., 2021). This dataset, derived from the Sentinel-2 mission of the Copernicus program, includes 27,000 labeled images with 10 to 60 meters spatial resolution, covering ten LULC classes across various regions of Europe.



Figure 1. EuroSAT dataset samples

These classes are Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River, and Sea/Lake (Helber et al., 2019). The EuroSAT dataset is available in two versions: true RGB images in JPG format and multispectral images in GeoTIFF format. The RGB version contains only the Blue, Green, and Red spectral bands. This makes it particularly suitable for computer vision models, as most deep learning architectures are inherently designed for three-channel data. On the other hand, the multispectral version provides all 13 spectral bands available in Sentinel-2 data, offering additional spectral information that may enhance classification accuracy. This study utilized both versions to assess the impact of additional spectral features on improving the

models' accuracy. Figure 1 shows sample EuroSAT images used in this study.

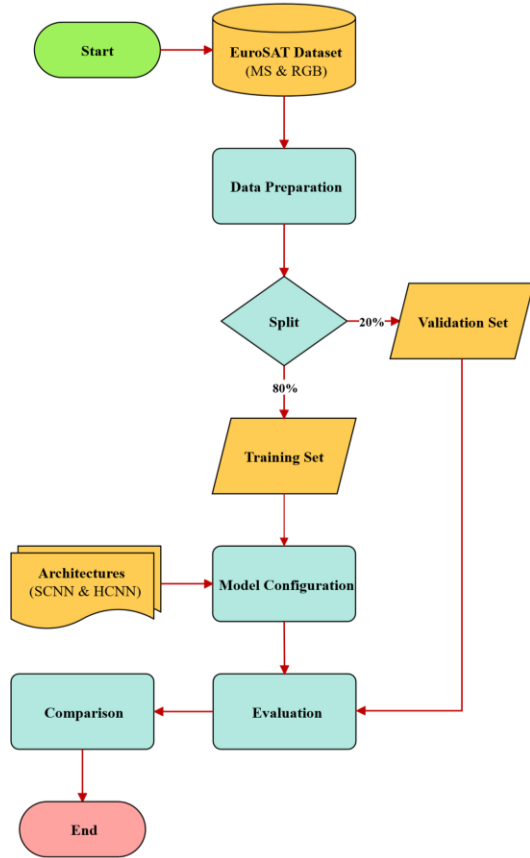


Figure 2. Methodology flowchart

2.2. Data Preparation

In this case, the EuroSAT dataset was divided into training and validation sets with an 80-20 split ratio. The ratio was applied separately to each LULC class rather than to the entire dataset. This ensured that the distribution of samples for each class was proportionally represented in both the training and validation sets. The process was performed randomly to avoid bias, ensuring a fair evaluation of the models. Table 1 shows the distribution of training and validation samples for each class. Finally, the digital numbers (DN) of the pixels were normalized using the Rasterio package.

Table 1. Training and validation set distribution (80:20)

ID	Class	Training Set	Validation Set
1	Annual Crop	2400	600
2	Forest	2400	600
3	Herbaceous Vegetation	2400	600
4	Highway	2000	500
5	Industrial	2000	500
6	Pasture	1600	400
7	Permanent Crop	2000	500
8	Residential	2400	600
9	River	2000	500
10	Sea/Lake	2400	600
-	Sum	21600	5400

2.3. Convolutional Neural Network Architectures

Two distinct architectures were designed and specifically developed for the objectives of this study. The first architecture has a simple structure similar to conventional CNN architectures. It was designed to achieve acceptable results while maintaining simplicity and avoiding excessive complexity. The second architecture was developed to improve accuracy and computational efficiency. It ensures that performance enhancements do not lead to increased complexity and reduced computational efficiency. The structural details of the architectures are elaborated in the following sections.

2.3.1 Single-Block Convolutional Neural Network

The single-block convolutional neural network (SCNN) consists of only one convolutional block. This architecture is designed to extract hierarchical spatial features through a structured sequence of convolutional, pooling, and fully connected layers. It begins with an initial convolutional layer with 64 filters and a 3×3 kernel to capture fundamental patterns. This is followed by batch normalization to stabilize learning and max-pooling to reduce spatial dimensions. As the network deepens, two additional convolutional layers with 128 and 256 filters, each using a 3×3 kernel, refine feature extraction. As a result, the model gains an improved ability to identify more complex patterns. Each convolutional layer is followed by a max-pooling layer to progressively down-sample the feature maps. A global average pooling (GAP) layer is then applied to condense spatial information into a compact representation. Finally, fully connected dense layers refine high-level features, incorporating a 512-unit dense layer with RELU activation and a dropout mechanism to prevent overfitting. Another 256-unit dense layer is added before the

final classification. The classification is then performed using a dense layer with softmax activation containing 10 output units, ensuring efficient and accurate results. The details of the SCNN architecture are presented in Table 2.

Table 2. The SCNN architecture summary

ID	Layer Type	Filters	Shape / Unit
1	Conv2D	64 (3×3)	64 × 64 × 64
2	MaxPooling2D	2×2	32 × 32 × 64
3	Conv2D	128 (3×3)	32 × 32 × 128
4	MaxPooling2D	2×2	16 × 16 × 128
5	Conv2D	256 (3×3)	16 × 16 × 256
6	MaxPooling2D	2×2	8 × 8 × 256
7	Global Average Pooling	-	192
8	Dense (ReLU)	-	256
9	Dropout	-	256
10	Dense (Softmax)	-	10

2.3.2 Hybrid Multi-Block Convolutional Neural Network

The Hybrid Multi-Block Convolutional Neural Network (HCNN) enhances feature extraction and classification performance by integrating multiple advanced architectural blocks. Similar to SCNN, this network begins with a convolutional layer with 64 filters and a 3×3 kernel for initial feature extraction. It is followed by batch normalization for stable learning and max-pooling to reduce spatial dimensions. A residual block with two convolutional layers, each with 64 filters and a 3×3 kernel, and a skip connection ensures efficient gradient flow. Next, dense blocks concatenate the outputs from three convolutional layers with 64, 128, and 192 filters, each using 3×3 kernels. This approach enhances feature reuse without applying batch normalization. A squeeze-excitation (SE) block dynamically recalibrates feature importance using global average pooling and two fully connected layers with 12 and 192 units. The output undergoes reshaping before multiplication with the feature map to emphasize key spatial features. This block reduces training time by compressing channel-wise information through global average pooling and using lightweight fully connected layers to minimize unnecessary computations. Another global average pooling layer condenses spatial information, reducing computational demand while preserving essential details. The classification stage consists of fully connected layers. It includes a dense layer with 512 units and ReLU activation, dropout regularization, and a final dense layer with 10 units and Softmax activation.

The integration of residual, dense, inception, and SE blocks into the base network architecture is expected to achieve a balanced trade-off between accuracy and computational efficiency. The details of the HCNN architecture are presented in Table 3.

Table 3. The HCNN architecture summary

ID	Layer Type	Filters	Shape / Unit
1	Conv2D	64 (3×3)	64 × 64 × 64
2	Batch Normalization	-	64 × 64 × 64
3	MaxPooling2D	2×2	32 × 32 × 64
4	Conv2D	64 (3×3)	32 × 32 × 64
5	Conv2D	64 (3×3)	32 × 32 × 64
6	Add (Skip Connection)	-	32 × 32 × 64
7	MaxPooling2D	2×2	16 × 16 × 64
8	Conv2D	64 (3×3)	16 × 16 × 64
9	Concatenate	-	16 × 16 × 64
10	Conv2D	128 (3×3)	16 × 16 × 128
11	Concatenate	-	16 × 16 × 128
12	Conv2D	192 (3×3)	16 × 16 × 192
13	Concatenate	-	16 × 16 × 192
14	Conv2D	256 (1×1)	16 × 16 × 256
15	Conv2D	256 (3×3)	16 × 16 × 256
16	Conv2D	256 (5×5)	16 × 16 × 256
17	Concatenate	-	16 × 16 × 192
18	Global Average Pooling	-	192
19	Dense	-	12
20	Dense	-	192
21	Reshape	-	1 × 1 × 192
22	Multiply Feature Maps	-	16 × 16 × 192
23	Global Average Pooling	-	192
24	Dense (ReLU)	-	512
25	Dropout	-	512
26	Dense (Softmax)	-	10

2.4. Model Configuration

For each HCNN and SCNN architecture, two models were trained based on RGB and multispectral data. Therefore, a total of four independent models were implemented for the objectives of this study. In the training process, the ADAM optimizer was used to control sudden gradient fluctuations and maintain the ideal convergence of the models. Thus, the initial learning rate of the models, set to 0.001, is adaptively adjusted and reconfigured based on gradient variations. In other words, when an increase in the gradient is detected, the learning rate is decreased accordingly, and when a decrease in the gradient occurs, the learning rate is increased accordingly. A maximum of 100 epochs was allocated for training the models. However, by adjusting the patience parameter, the overall performance of the models was monitored in each epoch using the Cross-Entropy loss function. This allowed the training process to be halted immediately after surpassing the patience threshold without a decrease in loss. This process not only optimizes the training process of the models but also plays a crucial role in monitoring potential overfitting and underfitting.

All implementation operations were performed on an RTX

3060 laptop GPU with 6 GB of Video Random Access Memory (VRAM). The TensorFlow package was used to enable GPU acceleration instead of relying on the Central Processing Unit (CPU). To facilitate direct reference in the results section, models were named SCNN-RGB, SCNN-MS, HCNN-RGB, and HCNN-MS, reflecting the architecture and the type of input data used.

2.5. Statistical Evaluation

In this study, various statistical metrics, including confusion matrices, overall accuracy (OA), kappa coefficient (KC), mean squared error (MSE), and root mean squared error (RMSE), were employed to evaluate the models and compare their results statistically. These evaluation approaches focus on assessing the accuracy and error rates of the models and examining the impact of architecture and input data on their statistical performance.

2.5.1 Confusion Matrix

A confusion matrix is a fundamental metric for evaluating classification models by comparing the classified samples with their corresponding actual labels. It is a square matrix where rows represent actual classes and columns represent classified classes. The diagonal elements indicate correctly classified instances, while off-diagonal elements represent misclassifications. Mathematically, it is structured as:

$$CM = \begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix} \quad (1)$$

where True Positive (TP) is correctly classified positive samples, False Positive (FP) is incorrectly classified positive samples, False Negative (FN) is incorrectly classified negative samples, and True Negative (TN) is correctly classified negative samples (Nicolau et al., 2024).

2.5.2 Overall Accuracy

Overall accuracy represents the ratio of correctly classified samples to the total number of samples. The overall accuracy is calculated as follows:

$$OA = \frac{\sum_{i=1}^k x_i}{N} \quad (2)$$

where x_i is the number of correctly classified samples for each class, k is the total number of classes, and N is the total number of samples used for assessment. Higher values indicate better classification performance. In remote

sensing applications, overall accuracy is commonly used to assess the accuracy of classification tasks, providing an overview of model performance (Congalton & Green, 2019).

2.5.3 Kappa Coefficient

The Kappa coefficient is a statistical measure used to evaluate the agreement between classifiers while accounting for random chance. Its values range from -1 to 1, where 1 indicates perfect agreement and 0 represents agreement expected by chance. Negative values suggest systematic disagreement, meaning the classifications are worse than random assignment. The kappa coefficient is calculated as follows:

$$kC = \frac{p_o - p_e}{1 - p_e} \quad (3)$$

where p_o is the overall accuracy, and p_e is the expected agreement by chance, derived from the confusion matrix (Cohen, 1960). This metric is widely used in evaluating classification maps using remote sensing data

2.5.4 Mean Squared Error and Root Mean Squared Error

Mean Squared Error (MSE) measures the average squared difference between actual and classified values, estimating the model's overall error. A lower MSE indicates a better model fit with fewer misclassifications. Mathematically, MSE is defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_j)^2 \quad (4)$$

where y_i represents the actual label, y_j is the classified sample, and n is the number of samples. Since MSE squares the differences, it is more sensitive to more significant errors, giving them a more significant influence on the final value. Root Mean Squared Error (RMSE), derived from MSE, is calculated by taking the square root of the MSE value:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_j)^2} \quad (5)$$

2.6. Computational Efficiency

The maximum time elapsed during the training process was recorded to assess the computational efficiency of the models. This evaluation provides a clearer understanding of how architectural differences affect computational speed. It also underscores the impact of data dimensionality, particularly the number of input channels, on this aspect.

2.7. Ablation Study

To assess the individual contribution of each block within the network structure, an ablation study was conducted on the HCNN architecture. As the HCNN architecture consists of residual, dense, inception, and squeeze-excitation blocks, unlike SCNN, each block was individually removed to evaluate its impact on the overall performance of the network. By comparing the results of the complete network architecture with those of versions where individual blocks were removed, the anticipated contribution of each block to accuracy and efficiency was identified.

3. Results

3.1. Statistical Evaluation Results

The trained models on RGB data, SCNN-RGB, and HCNN-RGB achieved overall accuracies of 94.19% and 95.54%, respectively. In contrast, SCNN-MS and HCNN-MS performed better in classifying correct samples than SCNN-RGB and HCNN-RGB. The SCNN-MS recorded an accuracy of 95.54%, while HCNN-MS achieved the highest performance among all models with an accuracy of 98.44%. The kappa coefficients for the SCNN-RGB, SCNN-MS, HCNN-RGB, and HCNN-MS were 93.53%, 97.04%, 95.03%, and 98.27%, respectively. This indicates the strong agreement between the classifications made by these models and the actual labels. The highest kappa coefficient value was associated with the HCNN-MS. The accuracy rates and kappa coefficient values of these models are presented in Figure 3.

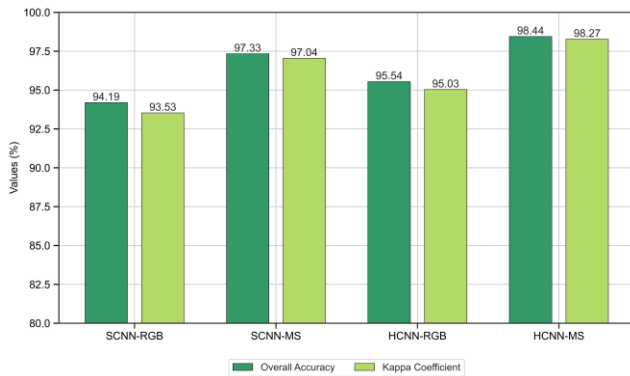


Figure 3. Accuracy and kappa coefficient graph

The error rates of the models, along with overall accuracy and kappa coefficient, are presented in Table 4. The HCNN-MS had the lowest error rate, whereas the SCNN-RGB had the highest. Based on all evaluation metrics, these models represent the highest and lowest performing approaches in this study. Although all four models demonstrated high accuracy, the HCNN-based models proved superior to those

with the SCNN architecture. Additionally, the type of input data had a significant impact on the performance of the models. In other words, the SCNN-MS and HCNN-MS, trained on multispectral data, outperformed those trained on RGB data due to their ability to extract a broader range of spectral features.

Table 4. Statistical evaluation values

Model	OA	KC	MSE	RMSE
SCNN-RGB	94.19	93.53	1.23	1.11
SCNN-MS	97.33	97.04	0.39	0.63
HCNN-RGB	95.54	95.03	0.77	0.88
HCNN-MS	98.44	98.27	0.21	0.46

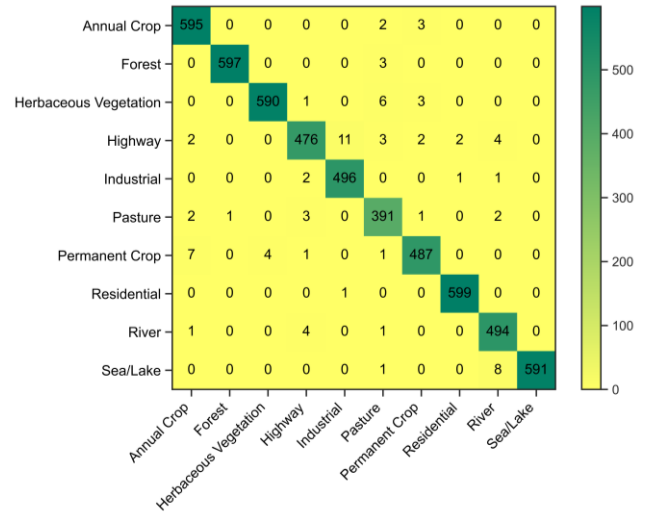


Figure 4. HCNN-MS confusion matrix

The analysis of the main diagonal of these matrices reveals more misclassifications in the SCNN-RGB and HCNN-RGB compared to the SCNN-MS and HCNN-MS. Figure 4 shows the confusion matrix of the HCNN-MS model, which was the most accurate model in this study. This difference in correctly classifying samples was minor in some classes but considerably more significant in others. For example, all models performed well in correctly classifying Sea/Lake samples. The SCNN-MS achieved 100% accuracy in this class without any misclassifications. Similarly, all models demonstrated high accuracy in classifying forest samples, with SCNN-MS achieving the highest accuracy of 99.8%. The models demonstrated varying performance in correctly classifying the samples of the Pasture class. SCNN-RGB, contrary to expectations, achieved the best performance in this class with an accuracy of 96.8%, outperforming the other models. The accuracy

Table 5. Block-wise ablation study of HCNN models

Model	Residual Block	Dense Block	Inception Block	SE Block	Accuracy (%)	Training Time (s)
HCNN-MS	✗	✓	✓	✓	94.17	815
	✓	✗	✓	✓	96.78	691
	✓	✓	✗	✓	96.69	651
	✓	✓	✓	✗	98.47	1906
	✗	✗	✗	✗	89.17	477
	✓	✓	✓	✓	98.44	977
HCNN-RGB	✗	✓	✓	✓	91.81	564
	✓	✗	✓	✓	93.26	461
	✓	✓	✗	✓	92.33	507
	✓	✓	✓	✗	95.89	1668
	✗	✗	✗	✗	82.88	381
	✓	✓	✓	✓	95.54	886

rate of the models in correctly classifying samples for each class is shown in Figure 5.

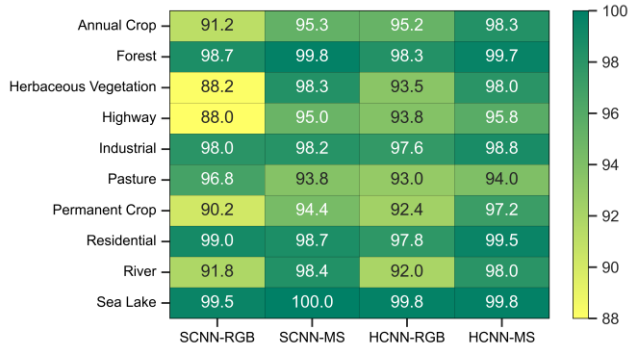


Figure 5. Accuracy reports for each class

3.1.1 Convergence Analysis

The convergence of the models was evaluated by analyzing the trend in the reduction of training and validation errors over 50 epochs. A consistent decrease in both metrics and a small gap between them indicates a balance between learning and generalization. All models demonstrated stable convergence, avoiding significant overfitting or underfitting. While some models exhibited a rapid initial decline in loss values, others showed a more gradual reduction before stabilizing. Minor fluctuations in validation loss were observed in some instances, suggesting slight sensitivity to validation data. However, loss trends confirm that the models effectively learned patterns in the data while maintaining generalization capability. Figure 6 shows loss trends.

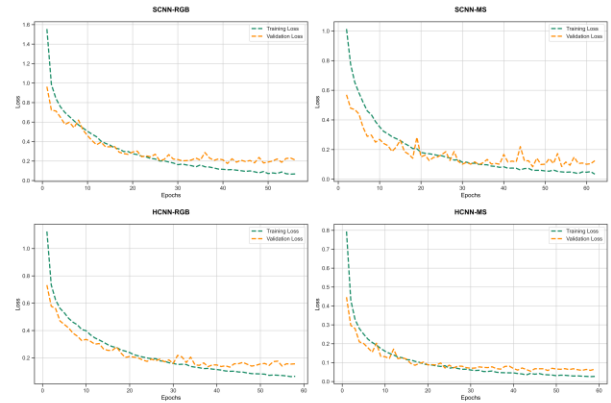


Figure 6. Loss trends graph

3.2. Computational Efficiency Analysis

The highest training time was observed for SCNN-MS, which required 6,344 seconds to complete. In comparison, SCNN-RGB completed training in 4,232 seconds, representing a 34% reduction in training time. In contrast, the HCNN-based models demonstrated significantly higher training efficiency, with HCNN-RGB and HCNN-MS completing in 886 and 977 seconds, respectively. Notably, HCNN-RGB required approximately 10% less training time than HCNN-MS. The 86% difference in training duration between HCNN-RGB and SCNN-MS highlights the

substantial variation in computational demand among the models.

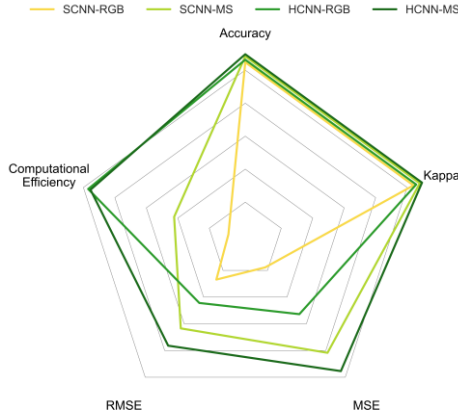


Figure 7. Overall performance of the models

Figure 7 shows the trade-off between statistical performance metrics and computational efficiency on a normalized scale. Although the models trained on multispectral data completed training in less time, the difference was more pronounced for the SCNN-based models than for the HCNN-based models. This indicates that the impact of the architecture on reducing complexity is far more significant than the impact of the dimensionality of input data.

3.3. Ablation Study Results

For HCNN-MS, removing the residual block reduced accuracy from 98.44% to 94.17% and slightly decreased training time from 977 to 815 seconds. Excluding the dense or inception blocks, accuracy dropped to 96.78% and 96.69%, with training times of 691 and 651 seconds, respectively. Removing the squeeze-excitation block had minimal impact on accuracy, slightly increasing it to 98.47%, but significantly increased training time to 1,906 seconds. Eliminating all blocks resulted in the lowest accuracy of 89.17% and the shortest training time of 477 seconds. Similarly, for HCNN-RGB, the absence of the residual block lowered accuracy from 95.54% to 91.81%, with training time reducing from 886 to 564 seconds. Excluding the dense and inception blocks reduced accuracy to 93.26% and 92.33%, with training times of 461 and 507 seconds, respectively. The removal of the squeeze-excitation block resulted in a marginal increase in accuracy to 95.89%, but significantly increased training time to 1,668 seconds. The model without any blocks achieved an accuracy of 82.88% and the shortest training time of 381 seconds. As shown in Table 5, each block contributes to

accuracy, especially the residual block. However, the squeeze-excitation block does not significantly affect model accuracy but plays a critical role in enhancing computational efficiency. As expected, this block compresses channel-wise information via global pooling and lightweight layers, reducing computational load and speeding up training. Consequently, its removal leads to increased training time.

4. Discussion

The results of this study showed that all four models trained on the EuroSAT dataset performed reasonably well in LULC classification. However, their overall performance varied depending on the network architecture and input data type. In general, trained models on RGB data exhibited higher classification errors and lower accuracy compared to models based on multispectral data. For example, the HCNN-MS achieved a significantly higher performance, reaching an accuracy of 98.44%, compared to 95.54% for the HCNN-RGB. The SCNN-based models exhibited similar behavior, highlighting the contribution of multispectral data to improved classification accuracy, particularly in distinguishing spectrally similar classes. Figure 8 provides a visual insight into the performance of the models in classifying 20 random samples.

Apart from the data type, the network architecture also played an essential role in reducing the computational complexity of the models. The HCNN architecture significantly reduced training time by optimizing the gradient flow and eliminating unnecessary computations. For instance, SCNN-MS required 6,344 seconds for training, while HCNN-MS completed training in 977 seconds, indicating a substantial decrease in computational time. In addition to reducing training time, HCNN-based models outperformed the other models in the evaluation criteria of this study, including overall accuracy and kappa coefficient. Among them, HCNN-MS achieved the best performance, followed by SCNN-MS, then HCNN-RGB, and finally SCNN-RGB. Therefore, it can be concluded that the HCNN-MS, due to its advanced architecture and the use of multispectral data, provided the best performance. In contrast, the SCNN-RGB, with a simpler architecture and limited data, showed the weakest performance in this study.

Table 6. Accuracy comparison of models in this study and other works

Model	Data	OA	Reference
SCNN *	RGB	94.19	This study
SCNN *	MS	97.33	This study
HCNN *	RGB	95.54	This study
HCNN *	MS	98.44	This study
VGG-19	RGB	77.82	Karaköse, 2024
ResNet-50	RGB	98.71	Panda et al., 2024
ResNet-101	RGB	99.01	Panda et al., 2024

GoogleNet	RGB	99.68	Panda et al., 2024
CNN	RGB	88	Acuña-Alonso et al., 2024
VGG-16	RGB	68	Adegun et al., 2023
DensNet-121	RGB	98	Adegun et al., 2023
CNN	MS	98.78	Yassine et al., 2021
CNN	MS+Indices	99.5	Yassine et al., 2021
CNN	RGB	96.83	Yassine et al., 2021
CNN	MS	94.9	Chong, 2020
VGG-16	RGB	94.5	Chong, 2020
RF	RGB	61.46	Sonune, 2020
VGG-19	RGB	97.66	Sonune, 2020
ResNet-50	RGB	94.25	Sonune, 2020

Compared to studies that have similarly used the EuroSAT dataset for LULC classification, different accuracies have been achieved depending on the network architecture and the proposed approach. For example, in one study, researchers achieved an accuracy of 97.66% using the VGG19 model, while another study using the same model reported an accuracy of 77.82%. Similarly, reported accuracies varied in studies utilizing the VGG16 model, with two studies achieving 94.5% and 68%, respectively. Table 6 shows that most of these studies used RGB data. Given the variety of CNN architectures used in these studies, the abbreviation 'CNN' is used in this table to represent them. The trained models in this study are distinguished by an asterisk (*) in the table. Considering these results, the architectures developed in this study are adequate, as all models achieved an accuracy exceeding 94%.

4.1. Limitations and Applicability

While the HCNN architecture demonstrated strong performance in LULC classification at the patch level, some limitations should be acknowledged to provide a balanced perspective. The architecture is specifically designed for patch-level classification, where each image patch is assigned, a single label based on its dominant LULC type. As such, it is not suitable for pixel-level or object-level classification tasks and cannot generate thematic maps that delineate detailed object boundaries. This limits its applicability in spatial segmentation or object detection scenarios. However, the architecture is highly flexible in terms of input data, supporting RGB, multispectral, and even single-band imagery with varying numbers of channels. The effectiveness of this architecture has only been evaluated on the EuroSAT dataset in this study. However, it can be applied to other datasets with similar characteristics, such as UC Merced and BigEarthNet, which also consist of labeled image patches. Therefore, one of the future directions of this research is to evaluate the architecture on more diverse and larger datasets to

overcome the geographical and contextual limitations of EuroSAT. Although its effectiveness has been primarily demonstrated in LULC classification, the architecture is not limited to this application and can be adapted for other patch-level classification tasks in remote sensing and general computer vision. While the proposed method is currently limited to patch-level classification, pixel-wise segmentation and thematic mapping are more critical in many practical remote sensing applications. Therefore, a key direction for future research could be to redesign the method within an encoder–decoder framework to enable pixel-wise segmentation, following the approach of architectures such as U-Net.

5. Conclusion

The advancement of satellite sensors has increased access to diverse and detailed data while also introducing challenges in processing high-dimensional inputs. In response, machine learning methods, particularly convolutional neural networks (CNNs), have become effective tools in satellite image processing and computer vision tasks. This study introduces a hybrid multi-block CNN (HCNN) architecture specifically designed to enhance accuracy and computational efficiency in LULC classification using the EuroSAT dataset derived from Sentinel-2 imagery. By integrating multiple architectural blocks alongside Batch Normalization and Global Average Pooling layers, the HCNN significantly outperformed a single-block (SCNN) across all statistical metrics. The HCNN models trained on RGB and multispectral data demonstrated both high accuracy and substantially reduced training time. Furthermore, the flexible structure of the HCNN enables compatibility with datasets of varying input dimensions, making it applicable not only to remote sensing but also to a broad range of computer vision tasks. The results are consistent with recent studies on deep learning for LULC classification using the EuroSAT dataset. In some cases, the proposed HCNN models achieve higher accuracy than common architectures, such as VGG16. Unlike many previous works, this study also emphasizes computational efficiency, a factor often overlooked in the related literature. Future research can build on this architecture by exploring more optimized variations that maintain high performance while reducing computational demands, thus facilitating deployment in low-resource hardware.

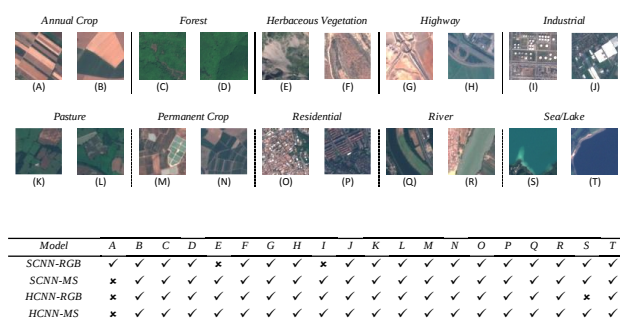


Figure 8. Classification results for 20 random samples

References

- Acuña-Alonso, C., García-Ontiyuelo, M., Barba-Barragáns, D., & Álvarez, X. (2024). Development of a convolutional neural network to accurately detect land use and land cover. *MethodsX*, 12, 102719. <https://doi.org/10.1016/j.mex.2024.102719>
- Adegun, A. A., Viriri, S., & Tapamo, J. R. (2023). Review of deep learning methods for remote sensing satellite images classification: experimental survey and comparative analysis. *Journal of Big Data*, 10(1), 93. <https://doi.org/10.1186/s40537-023-00772-x>
- Albarakati, H. M., Khan, M. A., Hamza, A., Khan, F., Kraiem, N., Jamel, L., ... & Alrooba, R. (2024). A novel deep learning architecture for agriculture land cover and land use classification from remote sensing images based on network-level fusion of self-attention architecture. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. <https://doi.org/10.1109/jstars.2024.3369950>
- Alyahyan, S. (2025). FusionNet remote a hybrid deep learning ensemble model for remote image classification in multispectral images. *Discover Computing*, 28(1), 3. <https://doi.org/10.1007/s10791-025-09498-1>
- Bhatti, M. K., Khan, M. A., Shaheen, S., Hamza, A., Arishi, A., AlHammedi, D. A., ... & Nam, Y. (2025). A Novel Approach for High-Resolution Coastal Areas and Land Use Recognition from Remote Sensing Images based on Multimodal Network-Level Fusion of SRAN3 and Lightweight Four Encoders ViT. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. <https://doi.org/10.1109/jstars.2025.3542194>
- Bing, Z. (2017). Current Status and Future Prospects of Remote Sensing. *Bulletin of Chinese Academy of Sciences (Chinese Version)*, 32(7), 774-784. <https://doi.org/10.11834/jrs.20166264>
- Blaschke, T. (2013, March). Object based image analysis: A new paradigm in remote sensing. In *ASPRS Annual Conference*, March (pp. 24-28).
- Calota, I., Faur, D., & Datcu, M. (2023). Dimensionality Reduction of Deep Learning for Earth Observation: Smaller, Faster, Simpler. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16, 4484-4498. <https://doi.org/10.1109/jstars.2023.3270384>
- Chen, G., Zhang, X., Tan, X., Cheng, Y., Dai, F., Zhu, K., ... & Wang, Q. (2018). Training small networks for scene classification of remote sensing images via knowledge distillation. *Remote Sensing*, 10(5), 719. <https://doi.org/10.3390/rs10050719>
- Chen, J., Huang, H., Peng, J., Zhu, J., Chen, L., Li, W., ... & Li, H. (2020). Convolution neural network architecture learning for remote sensing scene classification. *arXiv preprint arXiv:2001.09614*. <https://doi.org/10.48550/arXiv.2001.09614>
- Chong, E. EuroSAT Land Use and Land Cover Classification using Deep Learning. <https://github.com/e-chong/Remote-Sensing>, 2020.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1), 37-46. <https://doi.org/10.1177/001316446002000104>
- Congalton, R. G., & Green, K. (2019). Assessing the accuracy of remotely sensed data: principles and practices. CRC press. <https://doi.org/10.1201/9780429052729>
- Fayaz, M., Nam, J., Dang, L. M., Song, H. K., & Moon, H. (2024). Land-cover classification using deep learning with high-resolution remote-sensing imagery. *Applied Sciences*, 14(5), 1844. <https://doi.org/10.3390/app14051844>
- Filchev, L., & Chanev, M. (2024, June). Remote Sensing in Climate Change Research. In *The International Conference on Environmental Protection and Disaster Risks* (pp. 147-166). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-74707-6_17
- Ghimire, M. Remote Sensing and Image Analysis in Environmental Studies.
- Günen, M. A. (2022). Performance comparison of deep learning and machine learning methods in determining wetland water areas using EuroSAT dataset. *Environmental Science and Pollution Research*, 29(14), 21092-21106. <https://doi.org/10.1007/s11356-021-17177-z>
- Guo, F., Li, Z., Xin, Z., Zhu, X., Wang, L., & Zhang, J. (2021). Dual graph U-Nets for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14,

- 8160-8170.
<https://doi.org/10.1109/jstars.2021.3103744>
- Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7), 2217-2226.
<https://doi.org/10.1109/jstars.2019.2918242>
- Karaköse, E. (2024). An Efficient Satellite Images Classification Approach Based on Fuzzy Cognitive Map Integration with Deep Learning Models Using Improved Loss Function. *IEEE Access*.
<https://doi.org/10.1109/access.2024.3461871>
- Kunwar, S., & Ferdush, J. (2023). Mapping of land use and land cover (LULC) using EuroSAT and transfer learning. *arXiv preprint arXiv:2401.02424*.
<https://doi.org/10.48550/arXiv.2401.02424>
- Li, R., Gao, X., Shi, F., & Zhang, H. (2023). Scale effect of land cover classification from multi-resolution satellite remote sensing data. *Sensors*, 23(13), 6136.
<https://doi.org/10.3390/s23136136>
- Li, Y., Wang, L., Liu, X., Chu, Q., & Yang, X. (2021). A review of spatiotemporal super-resolution mapping for remote sensing data fusion. *IEEE Journal on Miniaturization for Air and Space Systems*, 3(1), 9-18.
<https://doi.org/10.1109/jmass.2021.3091837>
- Li, L., Zhang, W., Zhang, X., Emam, M., & Jing, W. (2023). Semi-supervised remote sensing image semantic segmentation method based on deep learning. *Electronics*, 12(2), 348.
<https://doi.org/10.3390/electronics12020348>
- Lian, Z., Zhan, Y., Zhang, W., Wang, Z., Liu, W., & Huang, X. (2025). Recent Advances in Deep Learning-Based Spatiotemporal Fusion Methods for Remote Sensing Images. *Sensors*, 25(4), 1093.
<https://doi.org/10.3390/s25041093>
- Lim, S. L., Sreevalsan - Nair, J., & Daya Sagar, B. S. (2024). Multispectral data mining: A focus on remote sensing satellite images. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 14(2), e1522.
<https://doi.org/10.1002/widm.1522>
- Liu, Q., Basu, S., Ganguly, S., Mukhopadhyay, S., DiBiano, R., Karki, M., & Nemani, R. (2020). Deepsat v2: feature augmented convolutional neural nets for satellite image classification. *Remote Sensing Letters*, 11(2), 156-165.
<https://doi.org/10.1080/2150704x.2019.1693071>
- Loganathan, A., Kousmitha, S., & Arun, Y. N. K. (2021, December). Land Use/Land Cover Classification Using Machine Learning and Deep Learning Algorithms for EuroSAT Dataset—A Review. In *International Conference on Intelligent Systems Design and Applications* (pp. 1363-1374). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-96308-8_126
- Lyons, M. B., Keith, D. A., Phinn, S. R., Mason, T. J., & Elith, J. (2018). A comparison of resampling methods for remote sensing classification and accuracy assessment. *Remote Sensing of Environment*, 208, 145-153.
<https://doi.org/10.1016/j.rse.2018.02.026>
- Jannat, T., & Hossain, M. A. (2024, May). Adapting VGG16 and ResNet50 for Cross-Domain Transfer Learning on Hyperspectral Image. In *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)* (pp. 1350-1355). IEEE.
<https://doi.org/10.1109/iceeict62016.2024.10534513>
- Jiang, M., Shen, H., Li, J., & Zhang, L. (2021). An Integrated Framework for the Heterogeneous Spatio-Spectral-Temporal Fusion of Remote Sensing Images. *arXiv preprint arXiv:2109.00400*.
<https://doi.org/10.48550/arXiv.2109.00400>
- Mohammed, E. A., & Lakizadeh, A. (2025). Benchmarking Vision Transformers for Satellite Image Classification based on Data Augmentation Techniques. *Int. J. Advance Soft Compu. Appl*, 16(3).
- Mohan, M., Macharla, A., P, P., Sharan, B., Nageswaran, A., & RM, B. (2024). Remote sensing-based ecosystem monitoring and disaster management in urban environments using machine learnings. *Remote Sensing in Earth Systems Sciences*, 7(4), 319-327.
<https://doi.org/10.1007/s41976-024-00124-0>
- Nicolau, A. P., Dyson, K., Saah, D., & Clinton, N. (2023). Accuracy assessment: quantifying classification quality. In *Cloud-Based Remote Sensing with Google Earth Engine: Fundamentals and Applications* (pp. 135-145). Cham: Springer International Publishing.
https://doi.org/10.1007/978-3-031-26588-4_7
- Niu, Y., Song, Z., Luo, Q., Chen, G., Ma, M., & Li, F. (2025). ATMformer: An Adaptive Token Merging Vision Transformer for Remote Sensing Image Scene Classification. *Remote Sensing*, 17(4), 660.
<https://doi.org/10.3390/rs17040660>
- Padmaja, S. M., Naveenkumar, R., Kumari, N. L., Pimo, E. S. J., Bindhu, M., Konduri, B., & Jangir, P. (2024). Deep Learning in Remote Sensing for Climate-Induced Disaster Resilience: A Comprehensive Interdisciplinary Approach. *Remote Sensing in Earth Systems Sciences*, 1-16.
<https://doi.org/10.1007/s41976-024-00178-0>

- Panda, S., Yadav, V. S., & Tripathi, V. K. (2024). Application of Remote Sensing in Natural Resource Management. In *Sustainable Development and Geospatial Technology: Volume 2: Applications and Future Directions* (pp. 173-180). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-65703-0_11
- Sharma, S., Beslity, J. O., Rustad, L., Shelby, L. J., Manos, P. T., Khanal, P., ... & Khanal, C. (2024). Remote Sensing and GIS in Natural Resource Management: Comparing tools and emphasizing the importance of in-situ data. *Remote Sensing*, 16(22), 4161. <https://doi.org/10.3390/rs16224161>
- Shi, C., Wang, T., & Wang, L. (2020). Branch feature fusion convolution network for remote sensing scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 5194-5210. <https://doi.org/10.1109/jstars.2020.3018307>
- Stival, L., da Silva Torres, R., & Pedrini, H. (2025). Semantically-Aware Contrastive Learning for multispectral remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 223, 173-187. <https://doi.org/10.1016/j.isprsjprs.2025.02.024>
- Sonune, N. Land Cover Classification with EuroSAT Dataset. <https://www.kaggle.com/nilesh789/land-cover-classification-with-eurosat-dataset>, 2020.
- Swain, R., Paul, A., & Behera, M. D. (2024). Spatio-temporal fusion methods for spectral remote sensing: A comprehensive technical review and comparative analysis. *Tropical Ecology*, 65(3), 356-375. <https://doi.org/10.1007/s42965-023-00318-5>
- Tiwari, P., & Shukla, P. (2018). Crop yield prediction by modified convolutional neural network and geographical indexes. *International Journal of Computer Sciences and Engineering*, 6(8), 503-513. <https://doi.org/10.26438/ijcse/v6i8.503513>
- Yassine, H., Tout, K., & Jaber, M. (2021). Improving lulc classification from satellite imagery using deep learning–eurosat dataset. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 369-376. <https://doi.org/10.5194/isprs-archives-xliii-b3-2021-369-2021>
- Yamashkin, S. A., Yamashkin, A. A., Zanozin, V. V., Radovanovic, M. M., & Barmin, A. N. (2020). Improving the efficiency of deep learning methods in remote sensing data analysis: geosystem approach. *IEEE Access*, 8, 179516-179529. <https://doi.org/10.1109/access.2020.3028030>
- Yuan, Z., Tang, C., Yang, A., Huang, W., & Chen, W. (2023). Few-shot remote sensing image scene classification based on metric learning and local descriptors. *Remote Sensing*, 15(3), 831. <https://doi.org/10.3390/rs15030831>
- Zeng, D., Chen, S., Chen, B., & Li, S. (2018). Improving remote sensing scene classification by integrating global-context and local-object features. *Remote Sensing*, 10(5), 734. <https://doi.org/10.3390/rs10050734>
- Zeng, Z., Wang, W., & Zhang, W. (2021, January). Target classification algorithms based on multispectral imaging: A review. In *Proceedings of the 2021 6th International Conference on Multimedia and Image Processing* (pp. 12-21). <https://doi.org/10.1145/3449388.3449393>
- Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE geoscience and remote sensing magazine*, 5(4), 8-36. <https://doi.org/10.1109/mgrs.2017.2762307>