SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I11P112

Volume 13 Issue 11, 194-220, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Class Complexity Score-Guided Curriculum Learning for Remote Sensing Land-Use Scene Classification

Kinjal A. Patel[1], Kaushika Patel[2]

[1]*Gujarat Technological University, Gujarat, India.*
[2]*Department of Electronics Engineering, BVM Engineering College, Gujarat, India*.

[2]*Corresponding Author: kdpatel@bvmengineering.ac.in*

Received: 28 August 2026 Revised: 29 September 2026 Accepted: 05 November 2026 Published: 27 November 2026

Abstract *- Remote sensing land-use scene classification suffers from high inter-class similarity and large intra-class diversity, yet standard training presents all classes together as if they were equally difficult to learn. This study introduces a novel, multi-signal Class Complexity Score (CCS)-guided curriculum learning framework that introduces entire classes in order of difficulty. CCS combines four signals, including inter-class feature similarity, prediction confusion score, prediction uncertainty, and intra-class feature variance from a trained baseline model. Classes are grouped into four difficulty bands as easy, moderate, hard, and very complex; moderate-to-very-complex classes receive class-specific RGB enhancement during training only. The model is then fine-tuned over four cumulative curriculum stages as per each difficulty band with progressive layer unfreezing. The framework was evaluated on the UC Merced dataset using a custom CNN trained from scratch and ImageNet-pretrained EfficientNetB0 and MobileNetV2. The curriculum improved all three backbones, achieving accuracies of 93.33%, 98.10%, and 97.46%, respectively, corresponding to gains of 5.40, 5.08, and 2.86 percentage points over their baselines. Wilcoxon, McNemar, and bootstrap analyses provided statistical support for the observed improvements. The research shows inter-class feature similarity plays a central role in maintaining the curriculum structure, while the remaining components provide complementary difficulty information.*

Keywords *- Remote sensing images, Land-Use scene classification, Curriculum learning, Class Complexity Score, Inter-Class diversity.*

1. Introduction

Remote Sensing (RS) acquires information about the Earth's surface without physical contact, using sensors on airborne and spaceborne platforms. Spaceborne RS now informs decisions on settlements, agriculture, water resources, forests, biodiversity conservation, health, disaster risk management, coastal zones, and energy [1]. Data from missions such as Landsat, MODIS, and Sentinel increasingly support monitoring of the United Nations Sustainable Development Goals (SDGs), including climate action, food security, water management, and life on land [2], and land cover-related variables underpin many SDG indicators as well as geographic information [3]. Land Use and Land Cover (LULC) classification is therefore central to most of these applications, because it turns raw imagery into maps of what occupies the land.

Integration of Artificial Intelligence (AI) and its advanced models plays a vital role in achieving SDGs and LULC classification [3, 4]. Figure 1 demonstrates the incorporation of fields such as RS, satellite imagery, Geographic Information System, along with AI to achieve SDGs. Deep learning has become the standard technique for LULC classification. Convolutional Neural Networks (CNNs) classify aerial and satellite scenes across benchmarks such as UC Merced, AID, NWPU-RESISC45, EuroSAT and SIRI-WHU, and regional LULC maps from Landsat and Sentinel imagery, along with Earth Observation [4, 6, 7]. Most of these methods fine-tune backbones pretrained on ImageNet [5], which carry millions of parameters designed for natural images. This raises memory and computation costs and is difficult to justify on small aerial datasets.

Recent work therefore pursues lighter models, such as EfficientNet-based ensembles [8], compact attention networks with about 2.5 M (million) parameters [9] and hybrid networks that cut parameters by 87% [10], but it reaches them by redesigning the architecture. Although the models are optimized and fine-tuned for LULC classification, two difficulties remain. First, misclassification concentrates among similar features of diverse classes, such as residential scenes of different densities, because aerial scenes have high inter-class similarity and large intra-class diversity [4, 11].

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

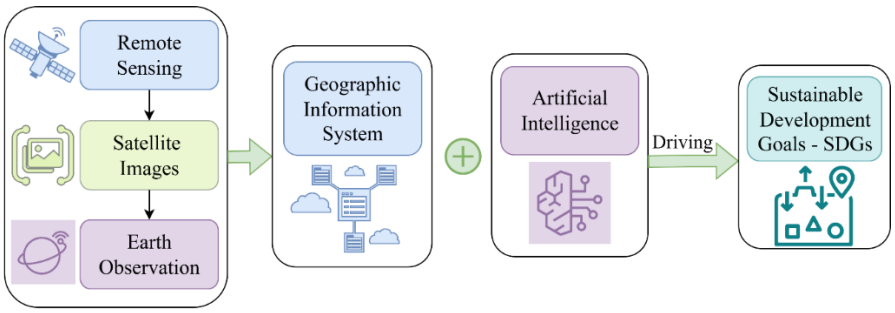

Fig. 1 Integration of RS with GIS and AI to achieve SDGs

Second, standard training presents all classes together in random order, ignoring that some classes are far harder to separate than others, curriculum learning, which trains on easier data before harder data [12], offers a way to exploit this difficulty without changing the architecture.

Although lightweight architectures and curriculum learning have been developed so far, the systematic use of class-level difficulty assessment and progressive learning has not been widely explored in remote sensing scene classification. Specifically, current strategies have not fully explored the interplay of inter-class similarity, prediction confusion, model uncertainty, and intra-class variance to define an architecture-specific learning sequence. To close this gap, the proposed framework includes these complementary indicators in a unified Class Complexity Score (CCS) that will be used for a progressive introduction of the classes based on the order of the various indicators.

The main contributions of the proposed framework are as follows:

1. Class-Level Curriculum Learning: Departs from predominantly sample-level curriculum learning approaches by progressively introducing entire remote sensing scene classes based on difficulty rather than ordering individual training samples.
2. Multi-Signal Class Complexity Score (CCS): Proposes a composite difficulty metric integrating four signals, including inter-class similarity, prediction confusion, model uncertainty, and intra-class variance, into a single interpretable score for class-level curriculum construction.
3. Backbone-Adaptive Difficulty Ordering: Establishes architecture-specific curriculum sequences using each backbone's learned feature representations and confusion patterns, rather than a fixed dataset-level difficulty ranking.
4. Class-Specific RGB Enhancement: Introduces a bank of six spectrally motivated RGB enhancement filters tailored to visually confusable scene categories, integrating original and enhanced images into curriculum-based training.
5. Curriculum-Coupled Progressive Unfreezing: Aligns architecture-specific network unfreezing with successive curriculum stages, progressively adapting model capacity as more difficult classes are introduced.
6. Lightweight Model Effectiveness: Achieves the largest absolute accuracy improvement of +5.40 Percentage Points (pp) using a custom CNN trained from scratch with only 1.74 M parameters, demonstrating the effectiveness of the proposed curriculum without reliance on large pretrained architectures.

The rest of this paper presents an account of the remainder in the following order. The relevant literature review of existing research studies is undertaken in Section 2, which leads to the gap to be addressed. Section 3 explains the desired curriculum learning methodology with CCS, and Section 4 presents the experimental setup. The curriculum learning results, cross-architecture comparisons, statistical validation, ablation study, and discussion are presented in Section 5. The current challenges and future directions are discussed in Section 6, and the paper is concluded in Section 7.

2. Literature Review of the Existing Approaches

Scene-level LULC classification has relied on a growing set of benchmarks. The UCM Land Use dataset was introduced by Yang and Newsam [13], who used it to evaluate bag-of-visual-words features and their spatial extensions. It contains 21 land-use classes with 100 aerial RGB images each; it remains among the most widely used benchmarks, alongside AID, NWPU-RESISC45, EuroSAT and SIRI-WHU [5]. Transfer learning has been the practical route for these small datasets. Hu et al. showed that ImageNet-pretrained CNNs without fine-tuning reach about 96.9% accuracy on UCM when their multi-scale convolutional features are encoded and classified with a linear SVM [14].

Albarakati et al. fused two custom deep networks with a self-attention layer and reported 98.20% on SIRI-WHU, 91.70% on NWPU-RESISC45, and 89.50% on EuroSAT [6]. Mashraqi et al. compared ten CNN-random forest hybrids, including VGG19, GoogleNet, DenseNet121, and ResNet, to map LULC in Najran, Saudi Arabia, from Landsat-8 imagery, and VGG19 with random forest reached 97.56% [7]. Deressu et al. used MobileNetV2, EfficientNet, ResNet and Xception encoders for LULC segmentation, with a best Dice coefficient of 91.3% [15]. Iman et al. reviewed deep transfer learning

more broadly and noted that freezing early layers is efficient but can bias a model toward its source domain, while full fine-tuning is prone to catastrophic forgetting [16].

The size of pretrained backbones has motivated work on lighter models. Huang et al. designed MS-DARNet, a 2.50 M-parameter attention Residual Network that reached 97.78% on AID [9]. Xu and Xu, fused CNN and transformer features through cosine-similarity cross-attention, reaching 97.81% on AID with an 87% parameter reduction [10]. Another study distilled a large network into a student with about 24 times fewer parameters than networks such as AlexNet [17]. Ashraf et al. found that their frequency-aware and texture-aware network performed best when trained from scratch, reaching 94.52% on UCM [18].

High inter-class similarity and large intra-class diversity are the defining challenges of aerial scene classification [4]. Chu et al. reviewed fine-grained recognition in RS, where subclasses within broad categories must be told apart [11]. Another study found, in a cross-domain setting with UCM as the target, that errors clustered on dense versus medium residential and on parking lot versus residential scenes, and they left inter-class similarity as an open limitation [19]. Xu et al. likewise named subtle inter-class similarities as a core challenge [10]. Farag et al. improved the separation of spectrally similar LULC classes with transfer-learned segmentation networks [20], and another study used a between-class similarity prior to guide a compressed network [17]. Outside RS, the study showed that one-vs-rest classifiers fail when classes are easily confused, and proposed an easy-to-hard paradigm that identifies easy classes automatically and uses them to help classify hard ones. Each of these works measures class difficulty with a single signal, if at all, and none uses it to schedule training [21].

Curriculum learning trains a model on easier material before harder material. Soviany et al. [12] surveyed the field, covering both predefined curricula and self-paced learning, in which difficulty is measured from the model's own loss. In RS scene classification, Liu et al. [22] paired a hierarchical vision transformer with a curriculum transfer-learning framework in which a difficulty evaluator schedules training samples. Bhat et al. [23] proposed CILEA-NET, which adds new land-cover classes to a trained CNN in an order based on their similarity to the classes already learned, to reduce forgetting in incremental learning. Other RS tasks use curricula at the sample or stage level: Cai et al. [24] reweighted object instances by difficulty for detection, Liu et al. [25] staged a semi-supervised segmentation pipeline that mixes poorly handled categories more often, and Rani et al. [26] applied a multi-level curriculum to YOLO models for crop growth-stage detection in precision agriculture. Although freezing protects pretrained knowledge while fine-tuning adapts it [16], none of these curricula ties the depth of fine-tuning to the difficulty of the classes being learned.

Preprocessing can also make classes easier to separate. Hung et al. [27] combined denoising, unsharp masking, Contrast-Limited Adaptive Histogram Equalisation (CLAHE) and Retinex methods with a CNN, raising test accuracy from 95.0% to 96.5% on RSSCN7 and from 97.68% to 98.01% on WHU-RS19, but the same enhancement was applied to every image. RGB-derived vegetation indices offer a physically motivated alternative for vegetated scenes. Bakacsy et al. [28] used six such indices on drone imagery to map invasive plants, and Moscovini et al. [29] predicted vegetation indices from calibrated RGB images with machine learning and deep learning. Li et al. [30] enhanced cloud-contaminated multispectral imagery with a MobileNet-based method. In all of these studies, enhancement is chosen per dataset or task rather than per class, and it is not linked to how hard each class is for the model.

2.1. Summary of Research Gaps

Recent approaches to UCM and related LULC datasets rely either on ImageNet-pretrained backbones used as feature extractors or fine-tuned networks [6, 7, 14, 15], or on lightweight architectures evaluated for accuracy and parameter count alone [8, 10, 17, 18]. None examines how a class-aware training procedure, independent of architecture, affects the accuracy-capacity relationship. In addition to that, the curriculum learning approaches reviewed in RS [22, 24-26] and in machine learning more broadly [12] are predominantly sample-level or stage-level. Class-level difficulty has been used mainly as a single-signal prior [17, 21], class-sequential training in RS serves incremental learning rather than difficulty ordering [23], and inter-class confusion remains an open limitation [11, 19, 20]. A composite, multi-signal per-class difficulty score used to order class introduction in fine-grained land-use classification has not been reported. Lastly, the reviewed enhancement strategies [27-30] are applied uniformly per dataset or task and are not coupled to model-derived per-class difficulty.

Among the studies reviewed, no approach was identified that jointly uses a composite, model-derived class difficulty score to guide both class-specific enhancement and progressive class introduction in remote sensing scene classification. Conventional end-to-end training strategies present all classes to the network in a fixed learning sequence without considering their relative levels of difficulty, thereby implicitly assuming that every class is equally challenging to learn.

However, land-use categories differ substantially in separability because of shared spectral characteristics, overlapping spatial structures, and varying degrees of intra-class variability in remote sensing imagery. Representative examples include the three residential-density categories and the runway-freeway-overpass group in the UC Merced dataset [13]. To address this limitation, the framework proposed in the following discussion addresses all gaps within a unified, class-level, difficulty-aware curriculum learning pipeline.

3. Proposed Research Work

The proposed framework and its related methodology phases are discussed, and Figure 2 shows the proposed workflow.

3.1. Methodology of the Proposed Framework

This study proposes a CCS-guided curriculum learning framework to improve fine-grained inter-class discrimination in satellite image classification. The proposed framework explicitly quantifies class-level learning difficulty and uses this information to identify classes that benefit from targeted spectral and textural enhancement and construct a curriculum that progressively introduces classes according to increasing difficulty while adapting the depth of network fine-tuning at each curriculum stage. The proposed framework comprises five sequential phases and is executed independently for three backbone architectures: a custom Convolutional Neural Network (CNN) trained from scratch, together with two ImageNet-pretrained transfer-learning models, EfficientNet-B0, and MobileNetV2.

1. Baseline training: Each backbone is first trained using the original data with a 70% training split.
2. CCS computation on the original dataset: The CCS is computed for each class by combining four complementary indicators: inter-class feature similarity, prediction confusion, prediction uncertainty, and intra-class feature variance using the trained baseline model, as discussed in Section 3.2.
3. CCS-guided semantic RGB enhancement: Based on their CCS values, classes are categorized into four difficulty bands such as Easy, Moderate, Hard, and Very Complex. Training images belonging to classes whose CCS values fall within the Moderate, Hard, and Very Complex difficulty bands are supplemented with class-group-specific enhanced versions while retaining their corresponding original images, resulting in a mixed training dataset that comprises both original and enhanced images. The validation and test splits remain unmodified, as discussed in Section 3.3.
4. CCS recomputation and difficulty-band curriculum construction: As a consistency check, CCS is recomputed once the mixed training dataset is created; the resulting class assignments define four cumulative curriculum stages, as discussed in Section 3.4.
5. Curriculum-based progressive fine-tuning: Beginning from the baseline model parameters, the network is fine-tuned sequentially across the cumulative curriculum stages, with progressively deeper network layers unfrozen as training advances, as discussed in Section 3.5.

As CCS is derived from the feature representation learned by each backbone rather than from a common architecture-independent metric, every backbone produces its own class-difficulty ranking, enhancement decisions, and curriculum composition. Consequently, a class identified as difficult by the custom CNN may not exhibit the same degree of difficulty within the feature space learned by EfficientNet-B0 or MobileNetV2. The curriculum is therefore generated independently for each backbone instead of being transferred across architectures.

3.2. Class Complexity Score (CCS) Formulation

A CCS is formulated to quantify class-level learning difficulty by integrating four complementary indicators derived from the behaviour of the trained baseline model on the validation split.

3.2.1. Inter-Class Feature Similarity (S)

Higher similarity values indicate greater overlap between class representations in the learned feature space and, consequently, a higher likelihood of inter-class confusion. For each class *i*, a centroid embedding c_i is obtained by averaging the feature vectors extracted from the network's final feature representation. The similarity score is defined as the maximum cosine similarity between the centroid of class *i* and the centroids of all remaining classes, as described in Equation 1:

$$S_i = \max_{j \neq i} \cos_{\text{sim}}(c_i, c_j) \tag{1}$$

3.2.2. Confusion Score (C)

The confusion score is computed from the baseline model's confusion matrix. Higher confusion scores indicate classes that are more difficult to distinguish from other categories. For each class *i*, it measures the normalized number of misclassification events involving that class as described in Equation 2:

$$C_i = \frac{\sum_j M_{ij} - M_{ii} + \sum_j M_{ji} - M_{ii}}{\max\left(\sum_j M_{ij'} 1\right)} \tag{2}$$

3.2.3. Prediction Uncertainty (U)

Although computationally efficient compared with Bayesian uncertainty estimation, this metric relies on softmax confidence. Higher values indicate lower model confidence. Prediction uncertainty is defined as one minus the maximum softmax probability assigned by the baseline model and is averaged over all samples belonging to class *i*, as described in Equation 3:

$$U_i = \text{mean}_{x \in \text{class } i} \left(1 - \max_k p(k \mid x)\right) \tag{3}$$

3.2.4. Intra-Class Feature Variance (V)

Higher variance indicates greater intra-class variability and increased learning difficulty. The intra-class variance score measures the dispersion of feature embeddings within each class by computing the mean Euclidean distance between each embedding and its class centroid, as described in Equation 4:

$$V_i = \text{mean}_{x \in \text{class } i} \parallel f(x) - c_i \parallel_2 \tag{4}$$

Each of the four component scores (S, C, U, and V) is independently normalized across all 21 classes using min-max normalization to obtain values in the interval $[0, 1]$. The final CCS for class i are then computed as the weighted aggregation, as described in Equations 5 and 6:

$$CCS_i = w_s\tilde{S}_i + w_c\tilde{C}_i + w_u\tilde{U}_i + w_v\tilde{V}_i \quad (5)$$

subject to

$$w_s + w_c + w_u + w_v = 1. \quad (6)$$

Where $\tilde{S}_i$, $\tilde{C}_i$, $\tilde{U}_i$, and $\tilde{V}_i$ denote the normalized inter-class feature similarity, confusion score, prediction uncertainty, and intra-class feature variance for class i, respectively.

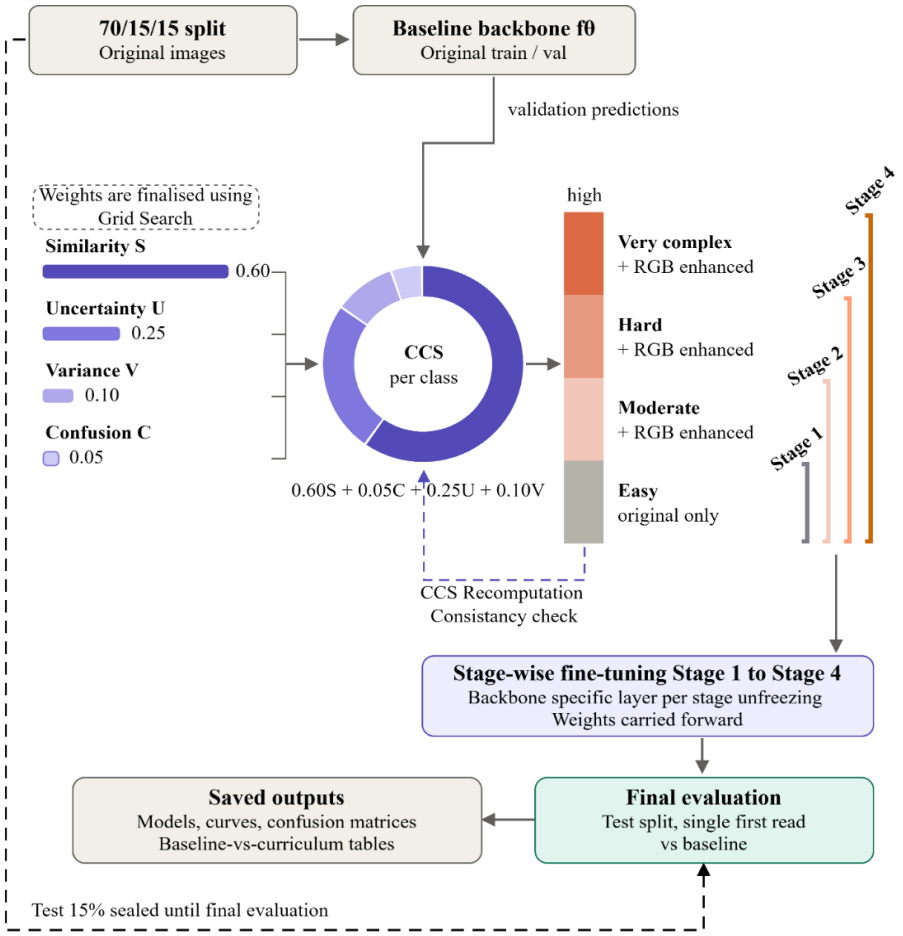

Fig. 2 Proposed workflow

The weights of the four CCS components were selected via grid search over multiple weight combinations, evaluated by curriculum learning performance.

The selected configuration Ws = 0.60, Wc = 0.05, Wu = 0.25, and Wv = 0.10 assigns the greatest importance to inter-class feature similarity, followed by prediction uncertainty, intra-class feature variance, and the confusion score.

3.3. CCS-Guided Semantic RGB Enhancement

Based on CCS computation, classes assigned to the Moderate, Hard, and Very Complex difficulty bands are selected for semantic RGB enhancement using the fixed thresholds reported in Table 1.

Table 1. Difficulty band to its corresponding CCS range

Difficulty band	CCS range
Easy	Below 0.35
Moderate	0.35 to below 0.55
Hard	0.55 to below 0.75
Very Complex	0.75 and above

Each selected class is mapped to one of six semantically motivated enhancement groups by its visual and physical characteristics discussed in Table 2**.** An asymmetric data protocol is used throughout the study, which ensures that curriculum gains cannot be attributed to an easier evaluation domain. Enhancement is applied exclusively to training set (original images together with enhanced images for CCS-flagged classes only) and to the validation set and test set original images only.

Table 2. Class-to-Enhancement-Filter mapping

Filter group	Classes	Enhancement logic
Vegetation Proxy	agricultural, forest, chaparral, golf course, baseball diamond	Excess Green Index (ExG = 2G − R − B) with a normalized-difference greenness measure related to NDVI: green-channel intensity increased, red channel suppressed, to improve vegetation/soil discrimination.
Water Proxy	beach, harbor, river	Blue/green channels enhanced, red attenuated, per water-likeness measure (B + G) / 2 - R, related to NDWI-based indices, exploiting water's characteristic spectral response.
Residential Proxy	dense residential, medium residential, sparse residential, mobile home park	CLAHE (clip limit 2.0, 8×8 tiles) on the LAB lightness channel, plus unsharp masking and Laplacian-based local variance enhancement, to emphasize rooftop density and structural differences.
Transport Proxy	freeway, overpass, runway, intersection	CLAHE combined with directional Sobel edge detection to strengthen axis-aligned lane markings and transportation boundaries.
Repetitive Pattern Proxy	parking lot, storage tanks, tennis court	Fine-grained CLAHE (clip limit 2.5, 4×4 tiles) with HSV saturation enhancement (×1.20) and mild unsharp masking, to emphasize repetitive patterns robustly to colour variation.
Industrial Proxy	buildings, airplane	CLAHE (clip limit 2.0, 8×8 tiles) with stronger HSV saturation enhancement (×1.25) and unsharp masking, to improve discrimination of metallic/concrete materials.

3.4. CCS Recomputation and Difficulty-Band Curriculum Construction

This recomputation will not affect the CCS values; it is only used to check that each class is in the correct difficulty band. The curriculum is then divided into four stages of learning, which are cumulative:

- Stage 1 (Easy): Includes only classes assigned to the Easy difficulty band.
- Stage 2 (Easy + Moderate): Expands Stage 1 by incorporating all Moderate classes.
- Stage 3 (Easy + Moderate + Hard): Extends Stage 2 through the addition of all Hard classes.
- Stage 4 (All Classes): Completes the curriculum by introducing the Very Complex classes, resulting in the full 21-class classification problem.

This cumulative structure allows the model to encounter increasingly difficult class boundaries while retaining all classes introduced during earlier stages.

3.5. Curriculum-based Progressive Fine-Tuning Strategy

Each backbone model is gradually refined with four curriculum stages from the saved baseline checkpoint, and the network parameters are not reinitialized between curriculum stages. A progressive layer-unfreezing strategy is used to reduce the disruption of limited-class training causing previously learned representations to be disrupted. This approach maintains old representations and gradually builds the deeper layers when adding more challenging classes. By Stage 4, all layers are trainable, enabling end-to-end optimization over the complete 21-class task. The corresponding architecture-specific schedules are summarized in the experimental setup section for each model. The held-out test split remains isolated throughout all intermediate stages and is accessed only once for final performance evaluation.

3.6. Algorithm Summary

Algorithm 1 summarizes the complete CCS-guided curriculum learning procedure.

Input: D_train, D_val, D_test (UCM 70/15/15 split)
backbone f_θ (CNN | MobileNetV2 | EfficientNet-B0)
weights $w = (w_s, w_c, w_u, w_v) = (0.60, 0.05, 0.25, 0.10)$
band thresholds (τ_{easy}, $\tau_{moderate}$, τ_{hard}) = (0.35, 0.55, 0.75)
stage epoch budget E = (E1, E2, E3, E4)

Output: curriculum-trained model f_θ^* and final test metrics
 # Phase 1 - Baseline training
1: $f_\theta \leftarrow$ train (f_θ, D_train, D_val)
2: (best checkpoint selected via D_val accuracy)

 # Phase 2 - CCS computation on the original dataset
3: for each class i: extract features $\varphi(x)$, compute centroid μ_i, and accumulate M
4: for each class i:
5: $S(i) \leftarrow \max_{j \neq i} \cos_sim(c_i, c_j)$
6: $C(i) \leftarrow$ (row+col errors for i in M) / support(i)
7: $U(i) \leftarrow$ mean (1 − max softmax prob) over class i
8: $V(i) \leftarrow$ mean $\|\varphi(x) - \mu_i\|_2$ over class i
9: $\tilde{S},\tilde{C},\tilde{U},\tilde{V} \leftarrow$ min-max normalize(S, C, U, V) over all i
10: for each class i: $CCS(i) \leftarrow w_s \cdot \tilde{S}(i) + w_c \cdot \tilde{C}(i) + w_u \cdot \tilde{U}(i) + w_v \cdot \tilde{V}(i)$

 # Phase 3 - Enhancement decision & mixed training-set construction
11: for each class i: band(i) ← threshold (CCS(i); τ_easy, τ_moderate, τ_hard)
12: for each class i with band(i) ≠ Easy:
13: g ← class_group(i)
14: D_train[i] ← D_train[i] ∪ apply_filter(D_train[i], filter_map[g])
15: D_train_mix ← resulting training set (original for all classes, + enhanced for flagged classes)

 # Phase 4 - CCS recomputation and difficulty-band curriculum
D_val and D_test remain original and unmodified throughout
16: repeat CCS computation on D_val using the unchanged baseline model to verify class assignments
17: B_easy, B_moderate, B_hard, B_verycomplex ← classes grouped by verified band
18: Stage_1 ← B_easy
19: Stage_2 ← Stage_1 ∪ B_moderate
20: Stage_3 ← Stage_2 ∪ B_hard
21: Stage_4 ← Stage_3 ∪ B_verycomplex

 # Phase 5 - Progressive curriculum fine-tuning
22: for stage k = 1 to 4:
23: configure_unfreezing(f_θ, backbone, k)
24: D_k ← {(x,y) ∈ D_train_mix : y ∈ Stage_k}
25: D_val_k ← {(x,y) ∈ D_val : y ∈ Stage_k}
26: f_θ ← fine_tune(f_θ, D_k, D_val_k)
27: f_θ* ← f_θ after Stage 4
28: final_metrics ← evaluate_once(f_θ*, D_test)
29: return f_θ*, final_metrics

4. Experimental Setup

The experiments were conducted in Python 3.10 with TensorFlow/Keras version 2.21. The image preprocessing and semantic RGB enhancement were achieved by OpenCV, Pillow, and NumPy, and scikit-learn was used for performance evaluation. The training was done on a single NVIDIA GeForce RTX 3070 Ti GPU (CUDA compute) with TensorFlow and GPU memory growth enabled. For the sake of reproducibility, a fixed random seed was used for dataset partitioning.

4.1. Dataset Description

The experiments were performed on the UCM Land Use dataset [13], which has 21 classes of land use with 100 images each and 2,100 images in total, where each image is an aerial RGB photograph of 256×256 pixels. Figure 3 is a demonstration of each class sample.

4.2. Data Preprocessing and Splitting

A fixed random seed of 42 was used for shuffling images for each class and splitting the images into training, validation, and test sets in a ratio of 70:15:15, across all 21 classes, giving 1470, 315, and 315 images in total, respectively. In all baseline and curriculum-learning experiments, the same fixed partition was employed, but in the curriculum-learning trials, the training partition was enhanced with CCS-guided enhanced images.

To enhance the model's generalizability without the need for extra labeled data, images were augmented in the training phase by randomly flipping them horizontally and vertically, rotating them by 0.15, zooming them by 0.15, and adjusting their contrast by 0.10.

4.3. Backbone Architectures and Baseline Training Configuration

Three backbone models were assessed to test the generality of the proposed framework under different model depths, capacities, and inductive biases. The detailed architectural configurations are presented in Tables 3-5 the corresponding network structures are shown in Figures 4-6.



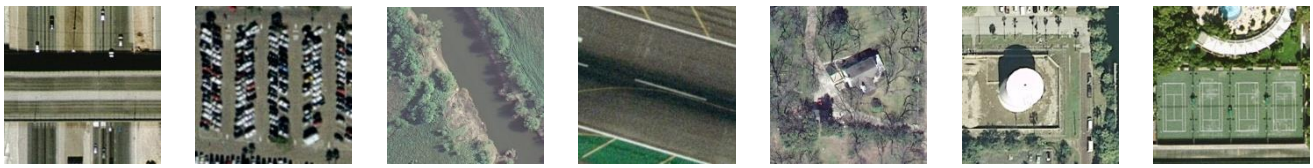

Fig. 3 UC merced Land Use dataset

The corresponding architecture is shown in Figure 4, where the sequential feature-extraction pipeline is depicted from the input image, via the convolutional blocks, the global average pooling, the fully connected classifier, and the 21-class softmax output.

Table 3 shows the detailed configuration of the custom CNN, consisting of 5 convolutional blocks, a classifier, output dimensions, and the number of parameters.

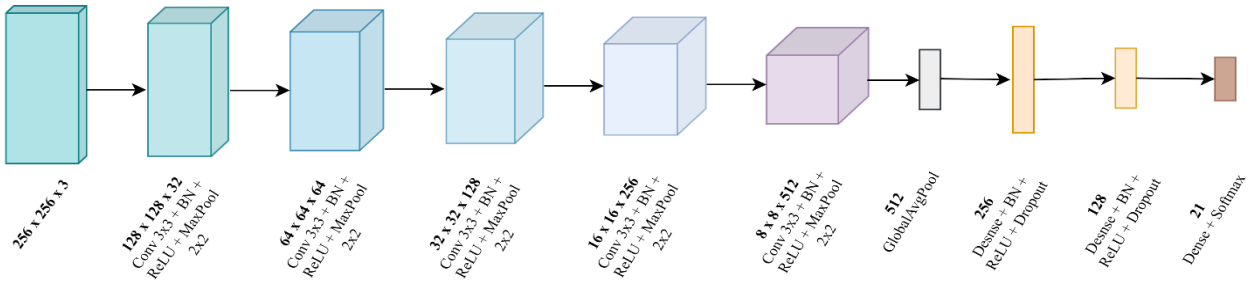

Fig. 4 Architecture of the custom CNN model

Table 3. Custom CNN baseline training configuration

Block	Operations	Kernel / Pool (size, stride, padding)	Input shape	Output shape	Parameters
Input	Rescaling (1/255 normalization)	-	256 x 256 x 3	256 x 256 x 3	0
Block 1	Conv2D (32) + BatchNorm + ReLU + MaxPool	Conv: 3 x 3 Pool: 2 x 2	256 x 256 x 3	128 x 128 x 32	1024
Block 2	Conv2D (64) + BatchNorm + ReLU + MaxPool	Conv: 3 x 3 Pool: 2 x 2	128 x 128 x 32	64 x 64 x 64	18,752
Block 3	Conv2D (128) + BatchNorm + ReLU + MaxPool	Conv: 3 x 3 Pool: 2 x 2	64 x 64 x 64	32 x 32 x 128	74,368
Block 4	Conv2D (256) + BatchNorm + ReLU + MaxPool	Conv: 3 x 3 Pool: 2 x 2	32 x 32 x 128	16 x 16 x 256	2,96,192
Block 5	Conv2D (512) + BatchNorm + ReLU + MaxPool	Conv: 3 x 3 Pool: 2 x 2	16 x 16 x 256	8 x 8 x 512	11,82,208
Pooling	GlobalAveragePooling2D	-	8 x 8 x 512	512	0
FC1	Dense (256) + BatchNorm + ReLU + Dropout (0.3)	-	512	256	1,32,352
FC2	Dense (128) + BatchNorm + ReLU + Dropout (0.3)	-	256	128	33,408
Output	Dense (21) + Softmax	-	128	21	2,709
Total					17,41,013 (17,38,261 trainable / 2,752 non-trainable, baseline)

As illustrated in Figure 5, the ImageNet-pretrained EfficientNet-B0 backbone serves as the feature extractor, followed by global average pooling, a 256-unit fully connected layer, dropout, and the final 21-class softmax classifier.

Table 4 details the EfficientNet-B0 configuration used in this study, including its MBConv stages, feature dimensions, parameter counts, and task-specific classifier head.

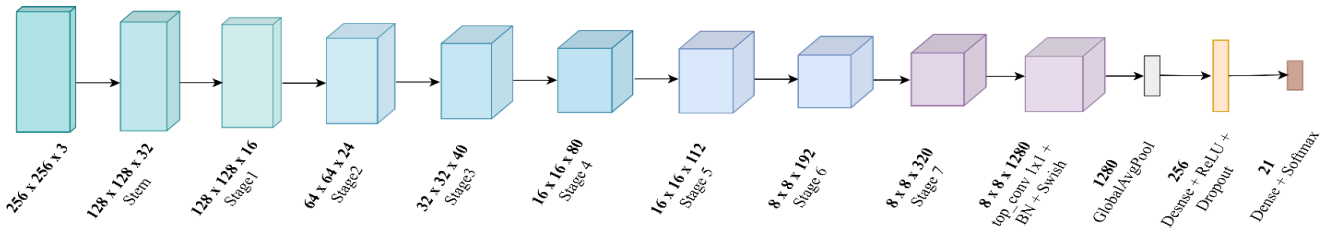

Fig. 5 Architecture of the custom EfficientNet-B0 model

Table 4. EfficientNet-B0 baseline training configuration

Stage	Block(s)	Cin	Expand	Kernel	Cout	Stride	Output shape	Params
-	Preprocessing (Rescaling + Normalization + Rescaling)	3	-	-	3	-	256 x 256 x 3	7
Stem	stem_conv + BatchNorm + Swish (SiLU) Activation Function	3	-	3 x 3	32	2	128 x 128 x 32	992
1	block1a (no expand + MBConv1	32	1x	3 x 3	16	1	128 x 128 x 16	1,544
2	block2a	16	6x	3 x 3	24	2	64 x 64 x 24	6,436
2	block2b (residual add)	24	6x	3 x 3	24	1	64 x 64 x 24	11,334
3	block3a	24	6x	5 x 5	40	2	32 x 32 x 40	16,006
3	block3b (residual add)	40	6x	5 x 5	40	1	32 x 32 x 40	32,330
4	block4a	40	6x	3 x 3	80	2	16 x 16 x 80	38,250
4	block4b (residual add)	80	6x	3 x 3	80	1	16 x 16 x 80	1,04,980
4	block4c (residual add)	80	6x	3 x 3	80	1	16 x 16 x 80	1,04,980
5	block5a	80	6x	5 x 5	112	1	16 x 16 x 112	1,28,148
5	block5b (residual add)	112	6x	5 x 5	112	1	16 x 16 x 112	2,11,484
5	block5c (residual add)	112	6x	5 x 5	112	1	16 x 16 x 112	2,11,484
6	block6a	112	6x	5 x 5	192	2	8 x 8 x 192	2,65,564
6	block6b (residual add)	192	6x	5 x 5	192	1	8 x 8 x 192	5,92,944
6	block6c (residual add)	192	6x	5 x 5	192	1	8 x 8 x 192	5,92,944
6	block6d (residual add)	192	6x	5 x 5	192	1	8 x 8 x 192	5,92,944
7	block7a	192	6x	3 x 3	320	1	8 x 8 x 320	7,22,480
Head	top_conv + BatchNorm + Swish (SiLU) Activation Function	320	-	1 x 1	1280	1	8 x 8 x 1280	4,14,720
Backbone Subtotal								40,49,571
Pooling	GlobalAveragePooling2D	-	-	-	-	-	1,280	0
FC1	Dense (256) + ReLU	-	-	-	-	-	256	3,27,936
-	Dropout (0.3)	-	-	-	-	-	256	0
Output	Dense (21) + Softmax	-	-	-	-	-	21	5,397
Grand Total								43,82,904 (3,33,333 trainable / 40,49,571 non-trainable)

The corresponding architecture is given in Figure 6, which involves feature extraction using the ImageNet-pretrained MobileNetV2 backbone, global average pooling, a fully connected layer of 256 units, dropout, and an output softmax layer with 21 classes. Table 5 shows the configuration of MobileNetV2, including the number of its inverted residual blocks, feature dimensions, number of parameters, and the classifier head.

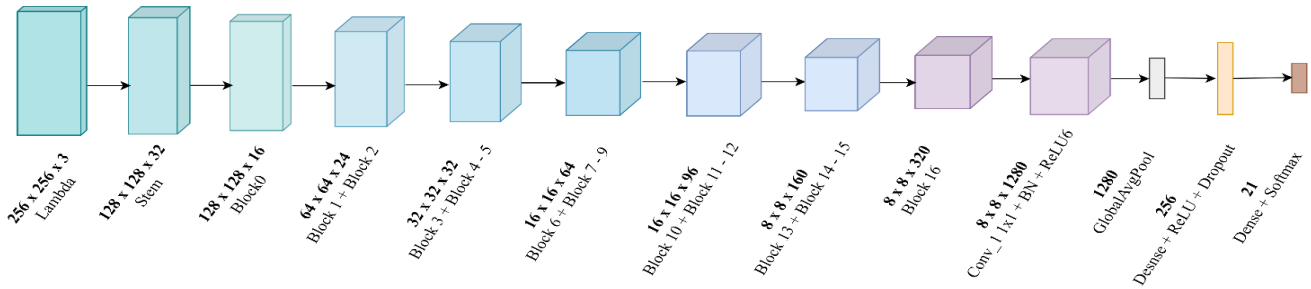

Fig. 6 Architecture of the MobileNetV2 model

Table 5. MobileNetV2 baseline training configuration

Stage	Block(s)	Cin	Expand	Cout	Stride	Output shape	Params
-	Preprocessing (Lambda)	3	-	3	-	256 x 256 x 3	0
stem	Conv1 + BatchNorm + ReLU6	3	-	32	2	128 x 128 x 32	992
0	Expanded_conv (no expand layer)	32	1x	16	1	128 x 128 x 16	992
1	block_1	16	6x	24	2	64 x 64 x 24	5,568
2	block_2 (residual add)	24	6x	24	1	64 x 64 x 24	9,456
3	block_3	24	6x	32	2	32 x 32 x 32	10,640
4	block_4 (residual add)	32	6x	32	1	32 x 32 x 32	15,680
5	block_5 (residual add)	32	6x	32	1	32 x 32 x 32	15,680
6	block_6	32	6x	64	1	16 x 16 x 64	21,952
7	block_7 (residual add)	64	6x	64	1	16 x 16 x 64	55,936
8	block_8 (residual add)	64	6x	64	1	16 x 16 x 64	55,936
9	block_9 (residual add)	64	6x	64	1	16 x 16 x 64	55,936
10	block_10 (channel change, add)	64	6x	64	1	16 x 16 x 96	68,352
11	block_11 (residual add)	96	6x	64	1	16 x 16 x 96	1,20,768
12	block_12 (residual add)	96	6x	64	1	16 x 16 x 96	1,20,768
13	block_13	96	6x	160	2	8 x 8 x 160	1,57,888
14	block_14 (residual add)	160	6x	160	1	8 x 8 x 160	3,24,160
15	block_15 (residual add)	160	6x	160	1	8 x 8 x 160	3,24,160
16	block_16 (channel change, no add)	160	6x	320	1	8 x 8 x 320	4,78,400
Head	Conv_1 + BatchNorm + ReLU (out_relu)	320	-	1280	1	8 x 8 x 1280	4,14,720
Backbone subtotal							22,57,984
Pooling	GlobalAveragePooling2D	-	-	-	-	1280	0
FC1	Dense (256) + ReLU	-	-	-	-	256	3,27,936
-	Dropout (0.3)	-	-	-	-	256	0
Output	Dense (21) + Softmax	-	-	-	-	21	5,397
Grand Total							25,91,317 (3,33,333 trainable / 22,57,984 non-trainable)

The number of epochs for training all three models was set to 200. The parameters of the best validation result on the model were used for baseline evaluation and as initialization for curriculum fine-tuning instead of the parameters after the final training epoch. Table 6 summarizes the optimization settings, the training loss, the learning rate, the epoch allocation, and the learning-rate scheduling strategy.

Table 6. Summary of key hyperparameters

Component	Setting	Baseline Training	Curriculum Training
Data	Image size	256 × 256 × 3	Same
	Batch size	32	32
	Random seed	42	42
Optimization	Optimizer	Adam	Adam
	Loss function	Sparse categorical cross-entropy	Sparse categorical cross-entropy
	Learning rate	0.001	0.001 (constant across all four stages)
	Maximum epochs	200	Total = 200 Stage 1 = 20, Stage 2 = 30, Stage 3 = 70, Stage 4 = 80

4.4. Curriculum Training Configuration

The curriculum learning was carried out in a modified 4-stage training schedule with a given number of epochs for each stage of training. A detailed description of the complete training configuration for both the baseline and curriculum learning is provided in Table 6.

For all backbones, the same learning rate was held for all learning phases. The number of classes added at each cumulative curriculum level varies slightly by architecture, as CCS is computed separately for each level. The outcomes of these curricular compositions for each backbone are summarized in Tables 7 and 8.

Table 7. CCS difficulty-band curriculum: Layers trained per stage, per backbone

Stage	Difficulty	Layers trained: Custom CNN	Layers trained: EfficientNet-B0	Layers trained: MobileNetV2
Stage 1	Easy	Classifier head[1] only; Convolutional blocks (1-5)[2] frozen	Classifier head[1] only; Entire EfficientNet-B0 backbone frozen	Classifier head[1] only; Entire MobileNetV2 backbone frozen
Stage 2	Easy + Moderate	Classifier head[1] unfrozen (including its own BatchNorm layers); convolutional blocks (1-5)[2] remain frozen	block7a[3] (7,22,480 params) + head[1] unfrozen	Final portion of block_16 + Conv_1[4] + head[1] unfrozen
Stage 3	Easy + Moderate + Hard	All layers unfrozen	block6a - block6d _ block7a[3] + head[1] unfrozen	Block_13 onward[4] + head[1] unfrozen
Stage 4	All classes	All layers unfrozen	All layers unfrozen, including top_conv (full fine-tune)	All layers unfrozen (full fine-tune)

[1]Classifier head: GlobalAveragePooling2D to Dense(256) to Dropout(0.3) to Dense(21, softmax), per Tables 3/4/5. For the custom CNN specifically, the head also includes two BatchNorm layers (after each Dense-256/Dense-128 layer); freezing is applied by layer type, so these are frozen along with the convolutional blocks at Stage 1 and unfrozen together with the rest of the head at Stage 2 - this is why Stage 2 for the CNN changes the trainable parameter count only slightly (166,933 to 168,469), not because a convolutional block is involved.

[2]Convolutional blocks 1-5: Five Conv2D+BatchNorm+ReLU+MaxPool blocks (32/64/128/256/512 filters), per Table 3.

[3]Block7a is EfficientNet-B0's deepest MBConv stage (722,480 params - the largest single block in the backbone); block6a-d are the four MBConv sub-blocks of Stage 6. The full per-stage breakdown is shown in Table 4.

[4]MobileNetV2's freeze code refers to these layers by fixed list position ("last 20 layers," "last 50 layers"), rather than by block name. The layer sequence and exact shapes are shown in Table 5.

Table 8. Per-Backbone curriculum stage-wise composition

Backbone	Stage 1 (Easy, 20 epochs)	Stage 2 (+ Moderate, 30 epochs)	Stage 3 (+ Hard, 70 epochs)	Stage 4 (+ Very Complex, 80 epochs)
Custom CNN	5	10	17	21
EfficientNet-B0	4	8	17	21
MobileNetV2	4	9	17	21

4.5. Evaluation Metrics

Model performance was evaluated on the held-out test set using overall accuracy, precision, recall, F1-score, Cohen's Kappa, and mean Average Precision (mAP). The same evaluation protocol was applied to the baseline models, the final curriculum-trained models, and the output of each curriculum stage.

Here, C denotes the total number of classes, N the number of test samples, and M_{ij} the confusion matrix entry representing samples of actual class i predicted as class j.

4.5.1. Accuracy

Percentage of samples correctly classified from all samples (all classes), as described in Equation 7.

$$\text{Accuracy} = \frac{\sum_{i=1}^{C} M_{ii}}{N} \quad (7)$$

4.5.2. Precision (Macro-Averaged)

Average of the proportion of correct positive predictions for all classes, as described in Equation 8.

$$\text{P}_i = \frac{M_{ii}}{\sum_{j=1}^{C} M_{ji}}, \text{Precision}_{\text{macro}} = \frac{1}{C}\sum_{i=1}^{C} \text{P}_i \quad (8)$$

4.5.3. Recall (Macro-Averaged)

Average percentage of correct responses for actual class samples, as described in Equation 9.

$$\text{R}_i = \frac{M_{ii}}{\sum_{j=1}^{C} M_{ij}}, \ \text{Recall}_{\text{macro}} = \frac{1}{C}\sum_{i=1}^{C} \text{R}_i \quad (9)$$

4.5.4. F1-Score (Macro-Averaged)

Average harmonic mean of precision and recall over all the Classes, as described in Equation 10.

$$F1_i = \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}, \ F1_{\text{macro}} = \frac{1}{C}\sum_{i=1}^{C} F\,1_i \quad (10)$$

4.5.5. Mean Average Precision (Macro, One-vs-Rest)

Measures the average precision–recall performance over all classes.

For each class i, treated as a binary one-vs-rest problem, Average Precision is the area under the precision-recall curve across confidence thresholds $\{t_n\}$ ranked by predicted probability, as described in Equation 11:

$$AP_i = \sum_n (R_n - R_{n-1}) \cdot P_n, \ \ mAP_{\text{macro}} = \frac{1}{C}\sum_{i=1}^{C} A\,P_i \quad (11)$$

Where P_n, R_n are precision and recall at threshold t_n.

4.5.6. Cohen's Kappa

Used to assess the level of agreement between predictions and classifications, taking chance agreement into account, as described in Equation 12.

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \ p_e = \frac{1}{N^2}\sum_{i=1}^{C}\left(\sum_{j=1}^{C} M_{ij}\right)\left(\sum_{j=1}^{C} M_{ji}\right) \quad (12)$$

where p_o = observed agreement = Accuracy, and p_e = expected agreement by chance, computed from the confusion matrix's row/column marginals:

4.6. Statistical Validation Framework

In addition to conventional performance metrics, statistical validation was performed to assess the significance and reliability of the proposed CCS-guided curriculum learning framework. Three pre-specified hypotheses were evaluated using a one-sided Wilcoxon signed-rank test (H1) and Spearman rank correlation (H2 and H3), with Holm–Bonferroni correction applied to control the family-wise error rate at $\alpha = 0.05$.

Supporting analyses included a directional exact McNemar test to compare paired image-level predictions and paired bootstrap resampling (5,000 iterations) to estimate 95% confidence intervals for improvements in accuracy, macro-precision, macro-recall, macro-F1, mean Average Precision (mAP), and Cohen's Kappa.

4.7. Ablation Study Design

A single-component removal ablation study was performed for the custom CNN backbone to assess the contribution of each ablation component. To obtain four variants of the ablation, the weight of one CCS component was set to zero while the weights of the other components were kept the same, as shown in Table 9.

Each ablation variant repeated the complete CCS framework, including CCS computation, enhancement-class selection, mixed training-set construction, difficulty-band assignment, and four-stage curriculum fine-tuning using its corresponding weight configuration. All remaining experimental settings were identical to those of the full model. Because CCS determines both enhancement selection and curriculum construction, removing an individual component may alter the resulting difficulty-band composition.

Table 9. Ablation variants and their (non-renormalized) CCS weights

Variant	Similarity (w_s)	Confusion (w_c)	Uncertainty (w_u)	Variance (w_v)	Weight sum
Full model	0.60	0.05	0.25	0.10	1.00
No-Similarity	0.00	0.05	0.25	0.10	0.40
No-Confusion	0.60	0.00	0.25	0.10	0.95
No-Uncertainty	0.60	0.05	0.00	0.10	0.75
No-Variance	0.60	0.05	0.25	0.00	0.90

5. Results and Discussion

The following results are obtained on the held-out test split (15 images per class, 315 images total per model) and using six different performance evaluation parameters, such as accuracy, precision, recall, F1-score, mean average precision, and cohen's kappa.

5.1. Overall Performance Comparison

The baseline models with their corresponding CCS-guided curriculum counterparts on the held-out original test set, and Table 10 compares all the models. The proposed framework consistently improves all three backbones.

The custom CNN outperforms, with classification accuracy rising from 87.94% to 93.33%, a 5.40 pp improvement.

EfficientNet-B0 achieves the highest overall classification accuracy, improving from 93.02% to 98.10%, with an improvement of 5.08 pp. Although MobileNetV2 already achieves a strong performance of 94.60%, it is further improved to 97.46%, which is a 2.86 pp gain. The proposed framework consistently enhances the overall classification performance, as shown by similar improvements across precision, recall, F1-score, Cohen's Kappa, and mAP, which are all evaluation metrics. The results are preliminary and show that the CCS-guided curriculum can help to learn features more effectively with both custom and pre-trained backbone architectures. An important observation is that the proposed CCS-guided curriculum allows the custom CNN to accomplish 93.33% accuracy, slightly better than the EfficientNet-B0 baseline accuracy of 93.02%.

Table 10. Result comparison - baseline models vs CCS-Guided curriculum learning models

Backbone and Method	Accuracy	Precision	Recall	F1-score	Cohen's κ	mAP	Accuracy Δ
Custom CNN Baseline	87.94%	87.91%	87.94%	87.69%	0.873	93.98%	-
Custom CNN CCS-Curriculum	**93.33%**	**93.60%**	**93.33%**	**93.31%**	**0.930**	**97.02%**	**+5.40 pp**
EfficientNet-B0 Baseline	93.02%	93.43%	93.02%	93.09%	0.927	97.79%	-
EfficientNet-B0 CCS-Curriculum	**98.10%**	**98.26%**	**98.10%**	**98.10%**	**0.980**	**99.80%**	**+5.08 pp**
MobileNetV2 Baseline	94.60%	95.13%	94.60%	94.70%	0.943	98.71%	-
MobileNetV2 CCS-Curriculum	**97.46%**	**97.69%**	**97.46%**	**97.44%**	**0.973**	**99.25%**	**+2.86 pp**

Considering that EfficientNet-B0 benefits from pretrained feature representations, this result highlights the effectiveness of the proposed framework in improving the performance of lightweight architectures trained from random initialization.

5.2. Learning Dynamics and Curriculum Progression

The overall learning dynamics were examined, as well as the learning development over curriculum stages, to gain a complete understanding of the proposed CCS-guided curriculum during training. First, the performance of the model was tracked at 10-epoch intervals during the entire 200 epochs of training using the conventional baseline training strategy and the proposed CCS-guided curriculum in Figures 7-9. The baselines tend to show a typical convergence trend, with a sharp rise in performance in the initial epochs and gradual leveling off over time. The CCS-guided curriculum, on the other hand, yields a clear multi-stage learning pathway, matching the introduction of increasingly difficult categories of scenes. Performance is observed to vary slightly between curriculum transitions, especially when Hard and Very Complex classes are added, but returns and reaches levels that are always higher than the corresponding baseline models by the end of training.

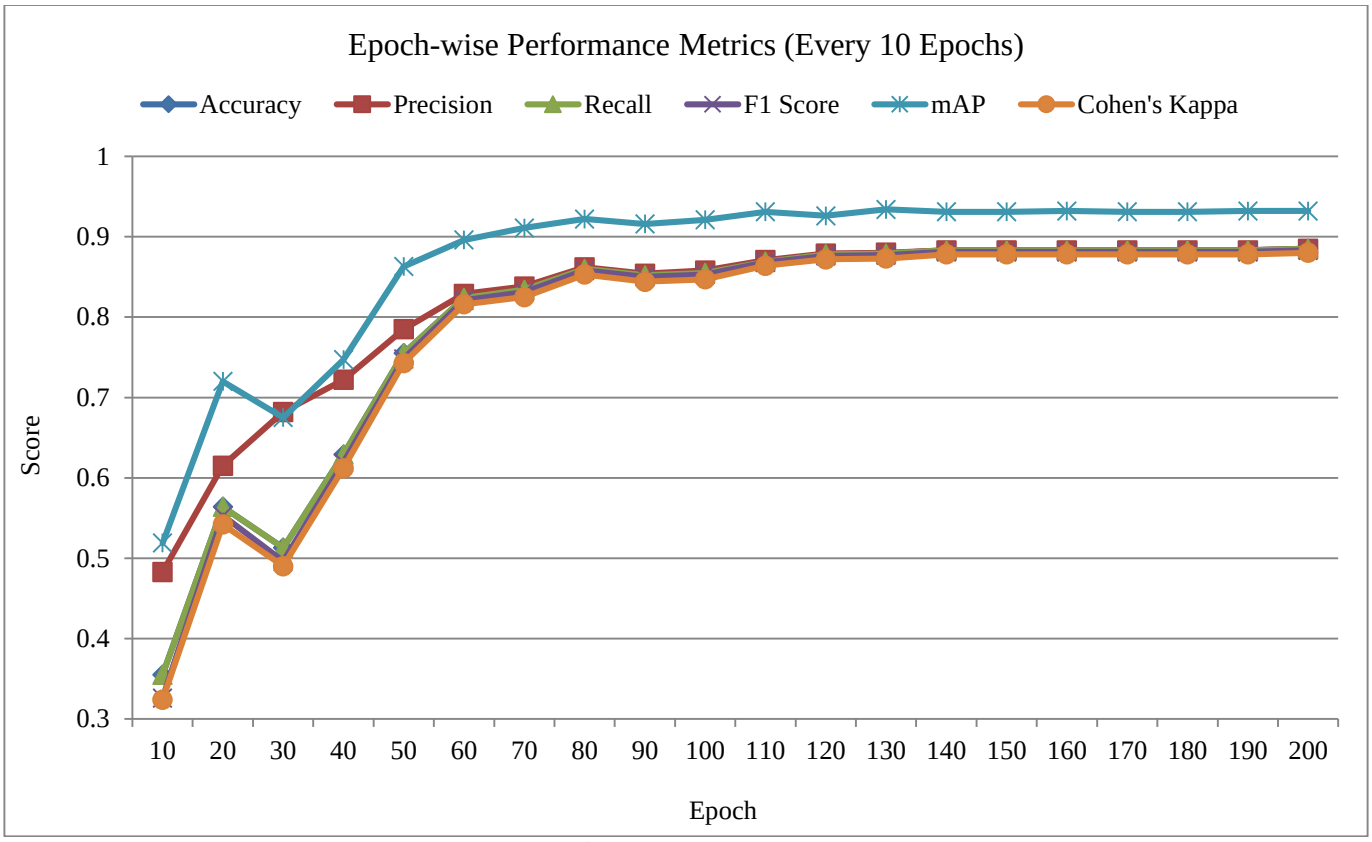

Fig. 7 (a) Epoch-wise evolution of the custom CNN under conventional baseline training

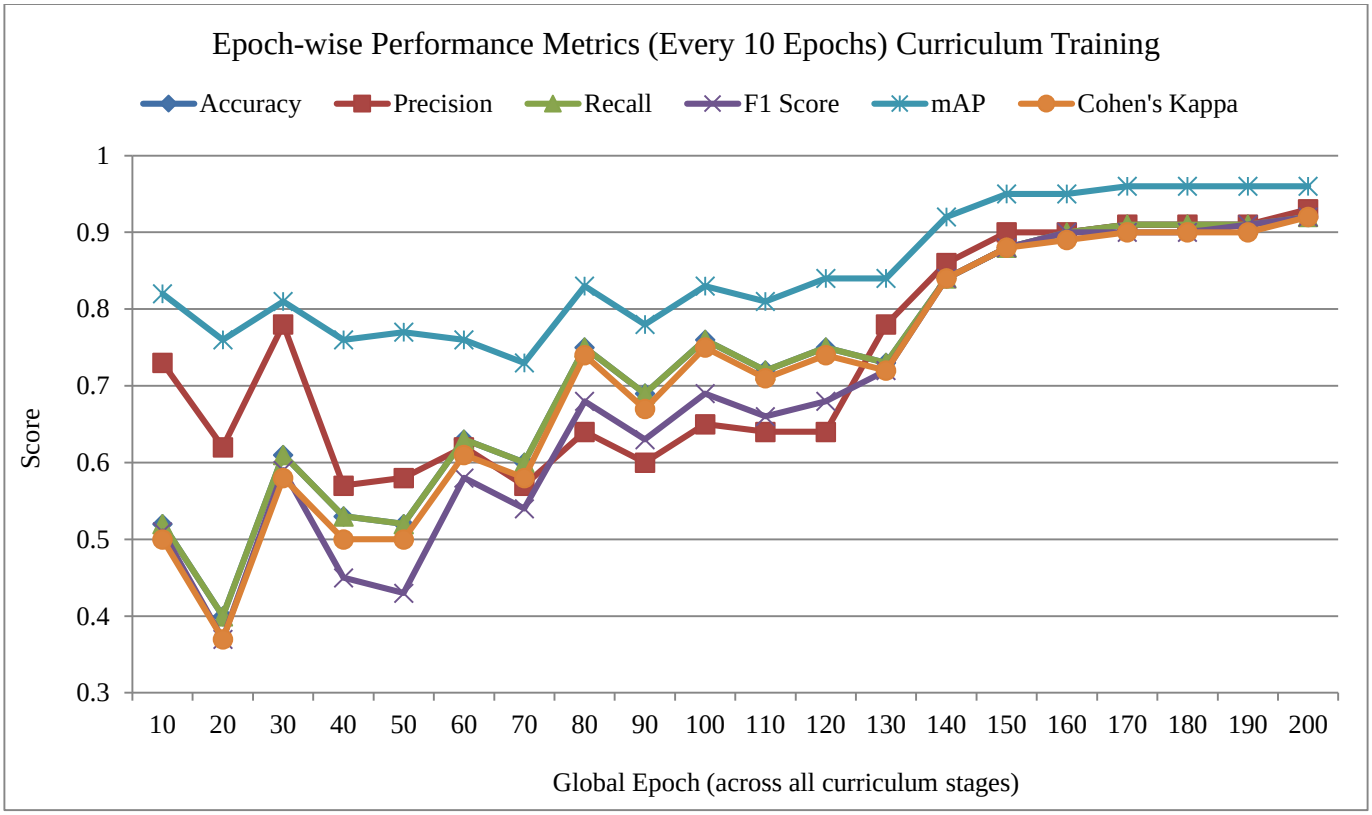

Fig. 7 (b) Epoch-wise evolution of the custom CNN under the proposed CCS-guided curriculum

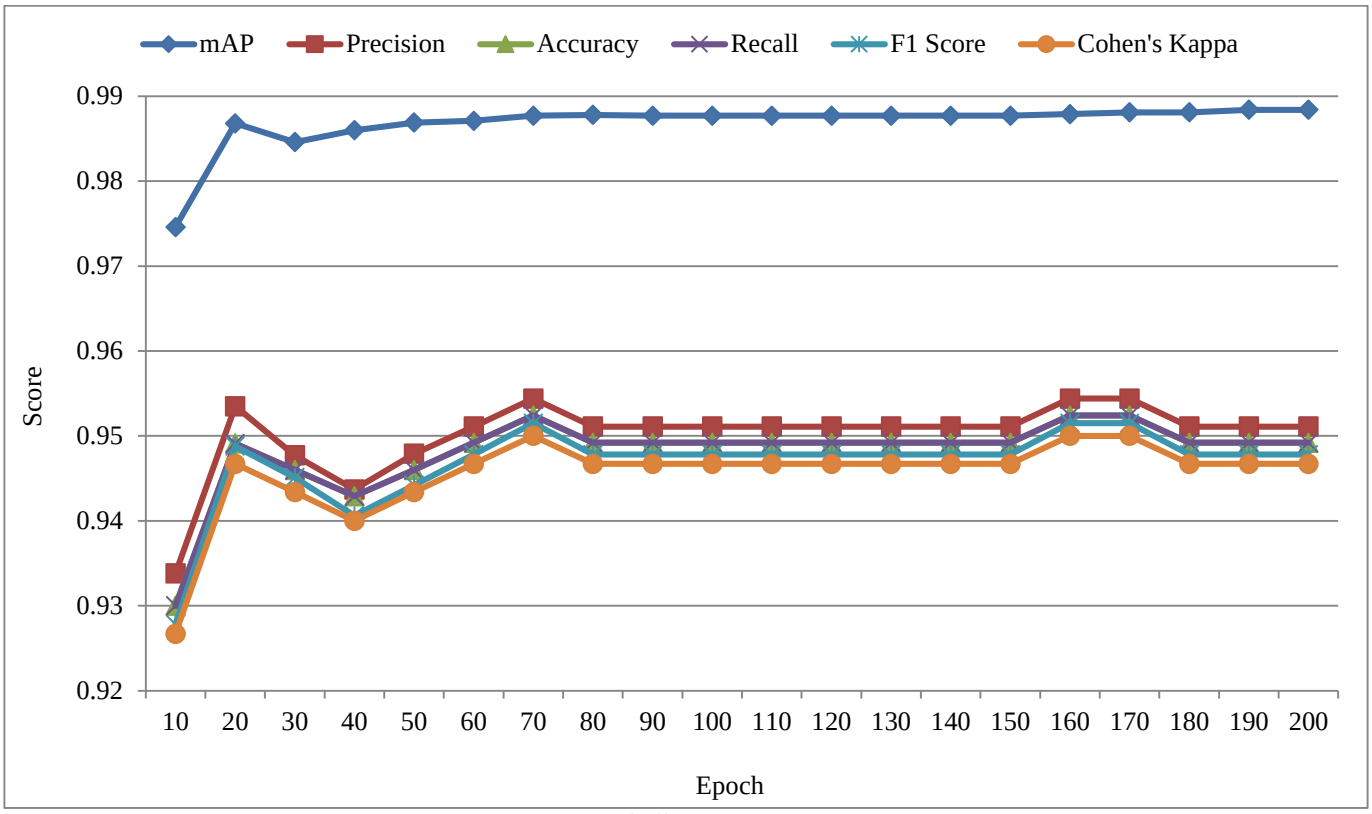

Fig. 8 (a) Epoch-wise evolution of EfficientNet-B0 under conventional baseline training

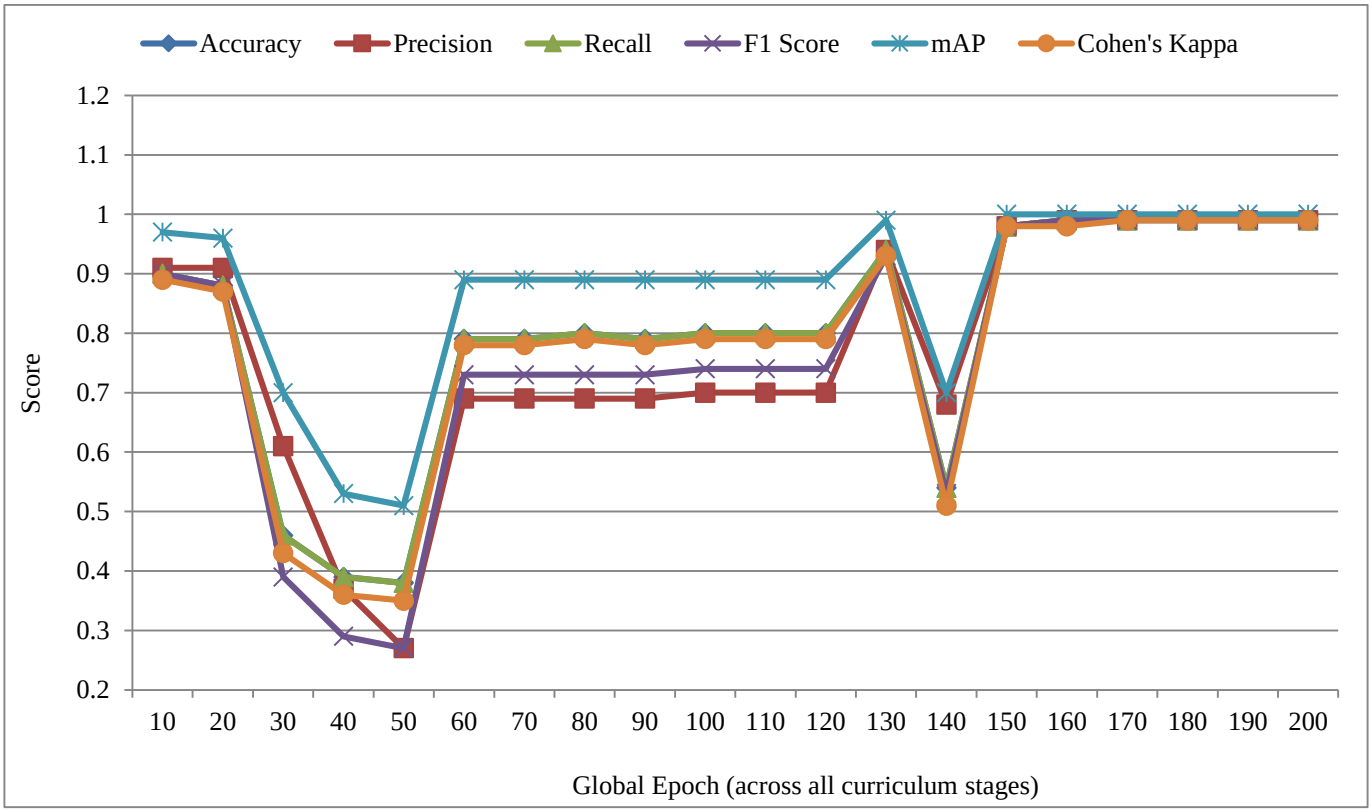

Fig. 8 (b) Epoch-wise evolution of EfficientNet-B0 under the proposed CCS-guided curriculum

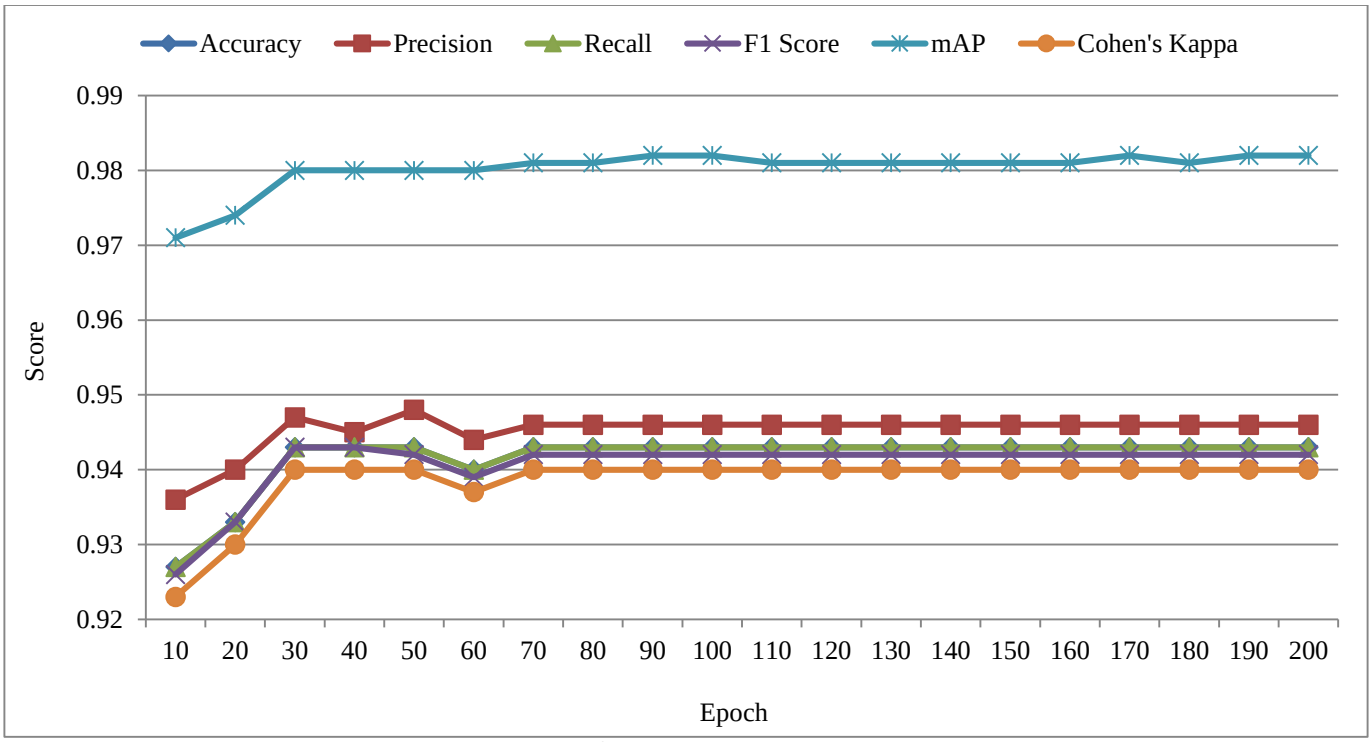

Fig. 9 (a) Epoch-wise evolution of MobileNetV2 under conventional baseline training

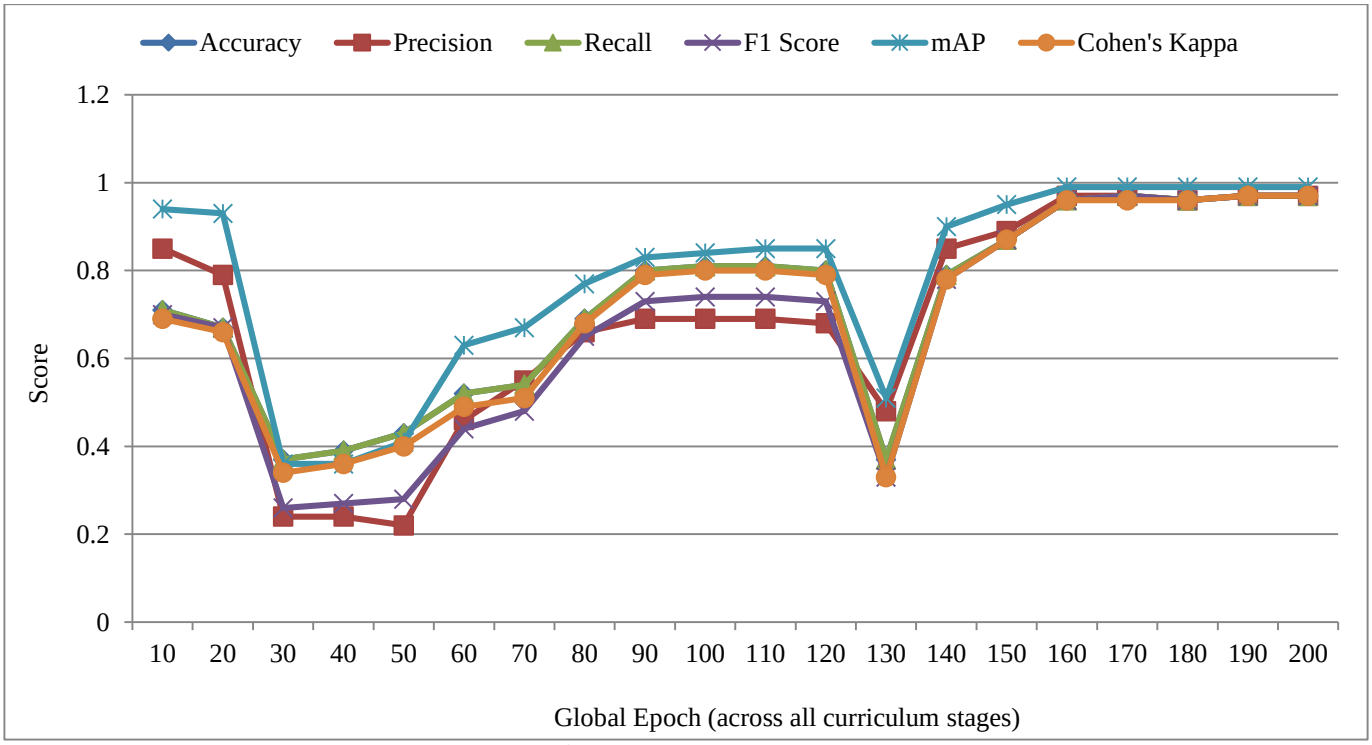

Fig. 9 (b) Epoch-wise evolution of MobileNetV2 under the proposed CCS-guided curriculum

As the overall improvement for the Custom CNN is the highest compared to the baseline, it means that the progressive curriculum learning is especially useful for the lightweight architectures that are trained from scratch. EfficientNet-B0 shows the largest performance dips at curriculum changes, but it quickly recovers and has the maximum classification accuracy at the end of the curriculum.

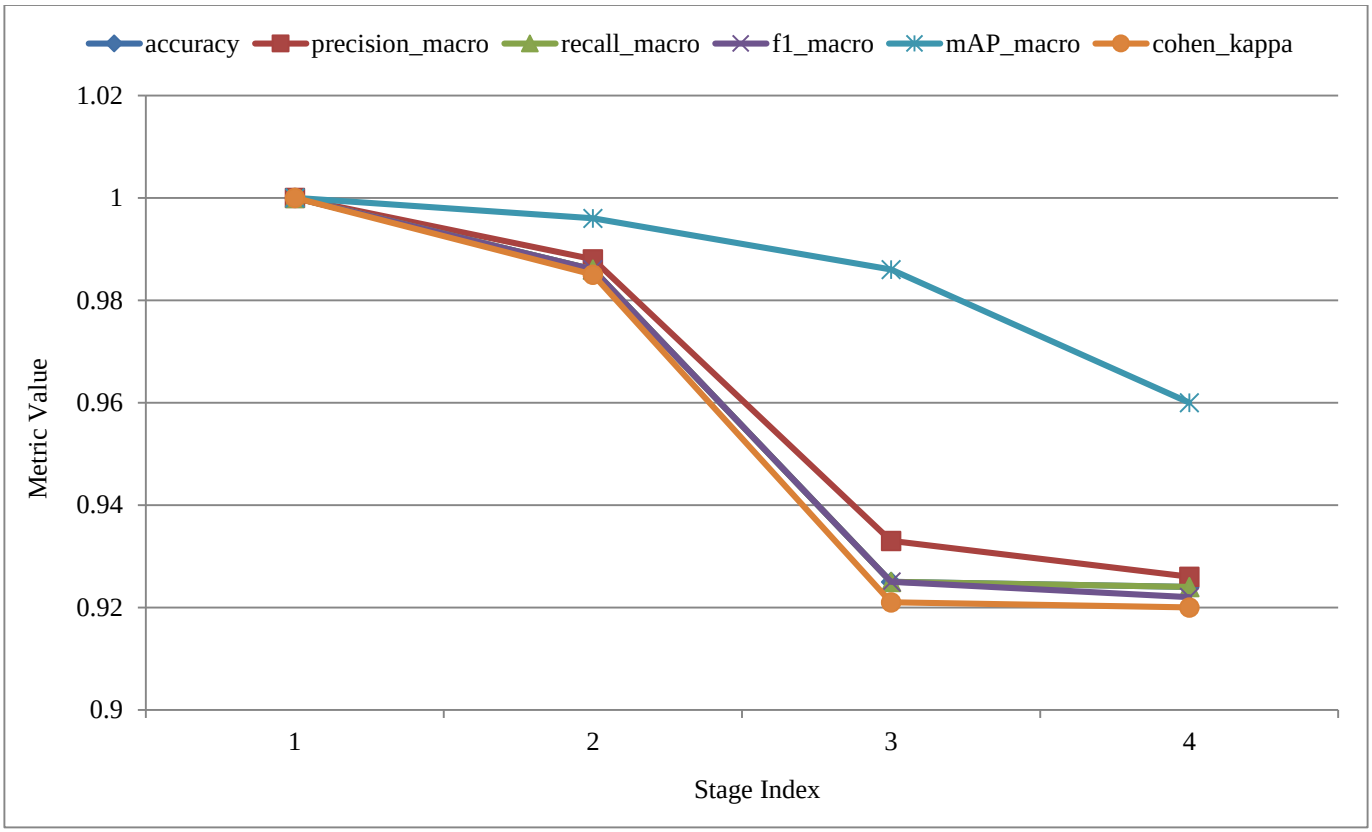

Fig. 10 Stage-wise validation performance of the custom CNN across the CCS-Guided curriculum

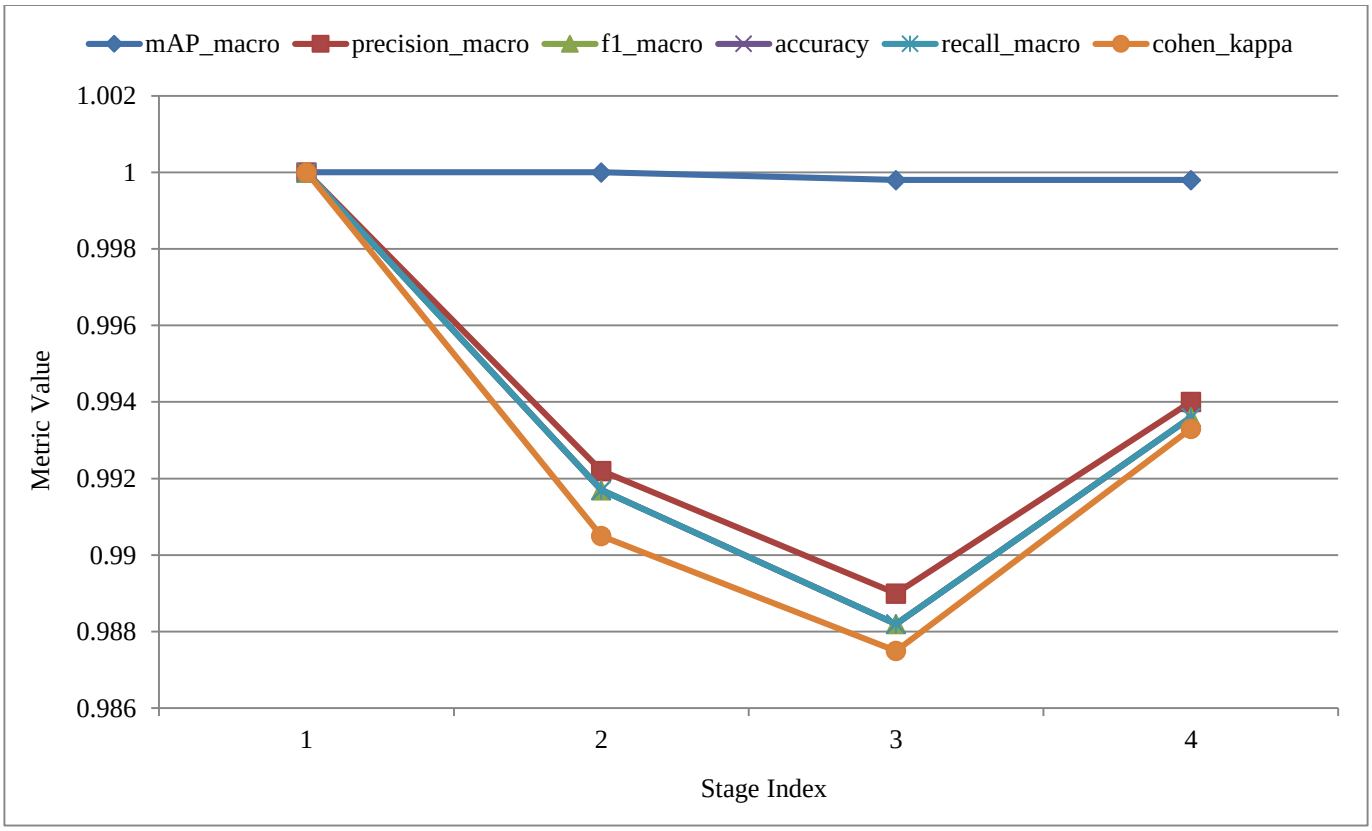

Fig. 11 Stage-wise validation performance of EfficientNet-B0 across the CCS-Guided curriculum

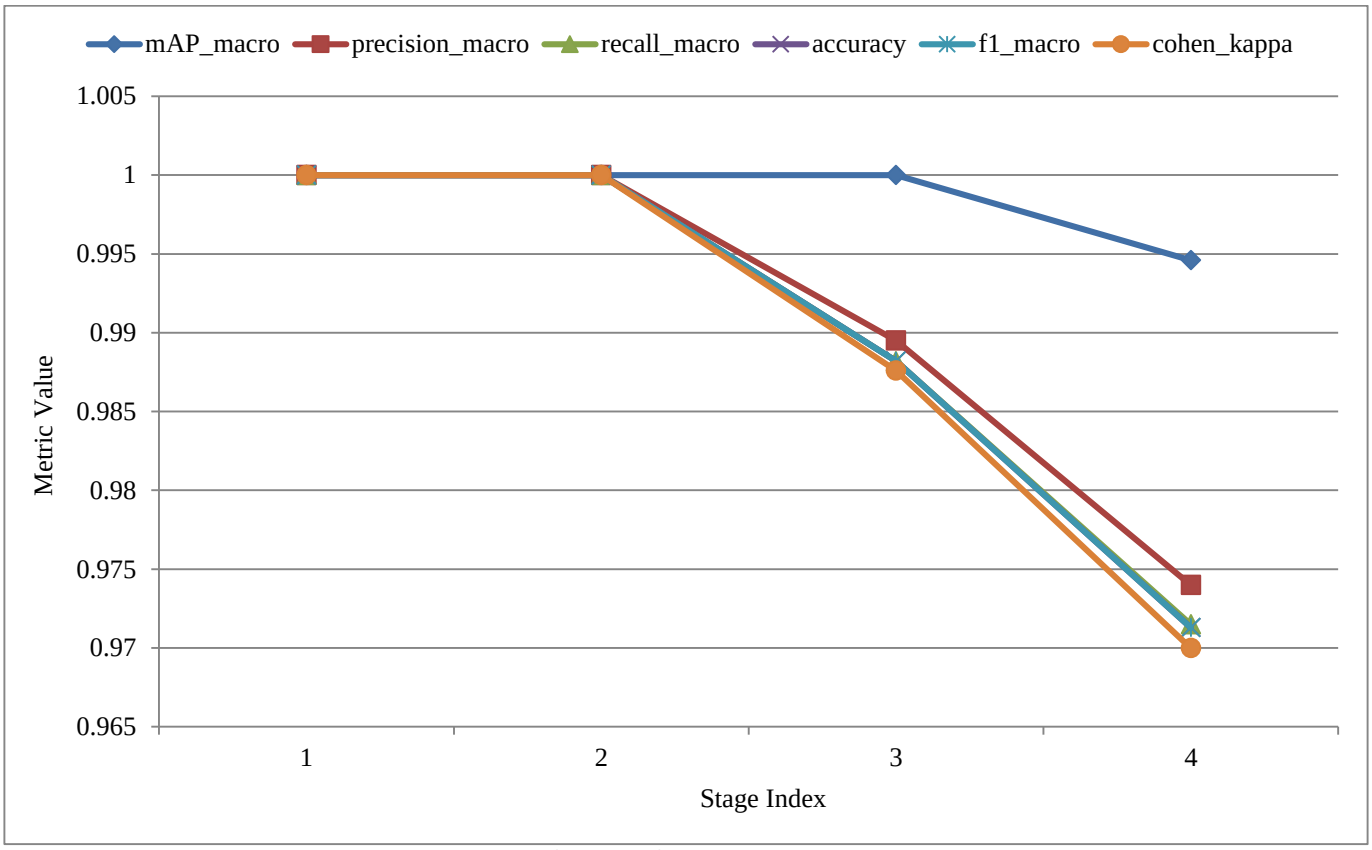

Fig. 12 Stage-wise validation performance of MobileNetV2 across the CCS-Guided curriculum

MobileNetV2 is relatively flat and has better performance than its base model in later stages of training. An interesting observation in all three backbones is that mAP stays consistently high relative to all other evaluation metrics across the training set as the scene classes are increasingly complicated, highlighting that the learned feature representations maintain good class-ranking capacity while training. Overall, the consistent improvement observed across six performance evaluation parameters, including Accuracy, Precision, Recall, F1-score, Cohen's Kappa, and mAP, demonstrates that the proposed CCS-guided curriculum enhances both optimization behaviour and final generalization performance across diverse backbone architectures. The four curriculum stages of each backbone architecture are further illustrated in Figures 10-12. However, mAP values typically stay relatively high over the course of the curriculum, especially for EfficientNet-B0 and MobileNetV2, while Accuracy, Precision, Recall, F1-score, and Cohen's Kappa tend to drop as increasingly difficult classes are added.

This indicates that the learned feature representations preserve strong class-ranking capability even as the classification task becomes increasingly difficult. Overall, the consistent stage-wise trends observed across all three backbone architectures are consistent with the intended curriculum design and provide visual evidence that increasingly challenging classes are introduced progressively throughout training.

5.3. Class-Level Performance Analysis

To further investigate the effect of the proposed CCS-guided curriculum, class-level performance was analysed by comparing the confusion matrices and per-class classification metrics of the baseline and curriculum-trained models. Figures 13-15 present the corresponding confusion matrices for the baseline and CCS-guided models evaluated on the held-out original-image test set.

The baseline CNN and the proposed CCS-guided curriculum are compared in Figure 13 in terms of their confusion matrices. The curriculum enhances or maintains recall for 18 of the 21 scene classes, showing that the performance gains observed are spread across a wide variety of scene classes. The largest improvements are obtained for tennis court (60.0% to 93.3%, +33.3 pp), overpass (66.7% to 93.3%, +26.6 pp), river (66.7% to 86.7%, +20.0 pp), and intersection (80.0% to 93.3%, +13.3 pp). A smaller increase (+6.7 pp) is noted in the agricultural, airplane, buildings, freeway, and medium residential categories, while only beach and sparse residential show a slight decrease (100.0% to 93.3%, -6.7 pp).

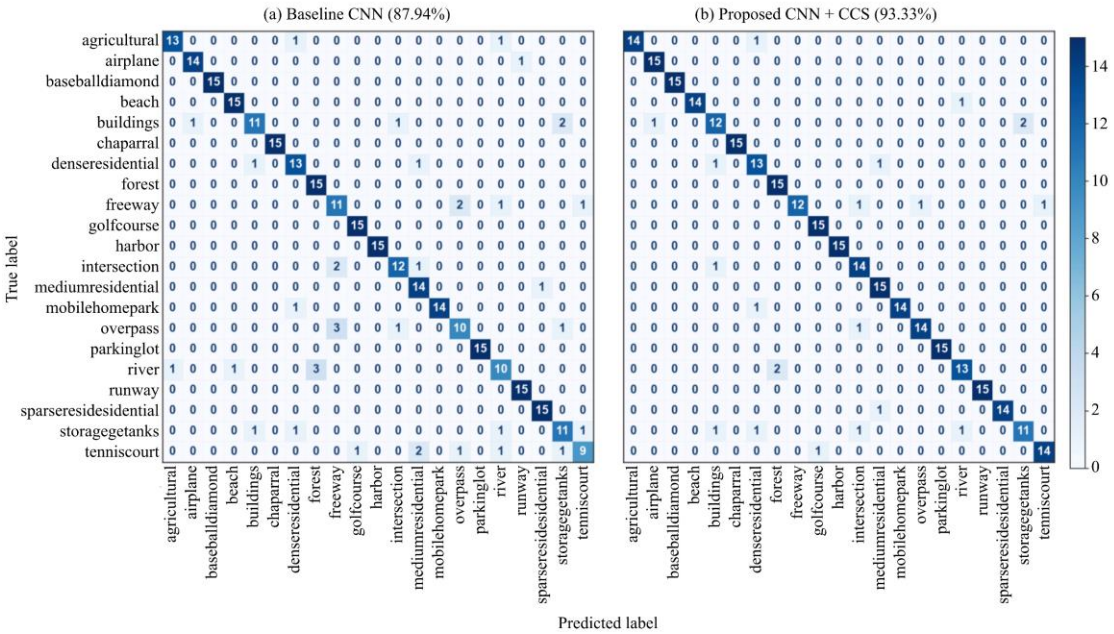

Fig. 13 Comparison of the confusion matrices obtained by (a) The baseline CNN, and (b) The proposed CCS-Guided curriculum model on the held-out original-image test set.

To compare the proposed CCS-guided curriculum with the baseline EfficientNet-B0, the confusion matrices are shown in Figure 14. The curriculum shows an improvement or maintained recall in 20 out of the 21 scene classes when compared to the baseline, indicating that the performance benefit is evenly spread throughout the dataset.The greatest progress is seen in the dense residential (+20.0 pp), intersection (+20.0 pp), medium residential (+20.0 pp), mobile home park (+20.0 pp), and tennis court (+13.3 pp). Smaller improvements are achieved for buildings, forest, and golf course (+6.7 pp each), while overpass exhibits a minor decrease from 93.3% to 86.7% (-6.7 pp).

The confusion matrices further show a stronger concentration of predictions along the principal diagonal, indicating reduced inter-class confusion and improved discrimination of visually similar scene categories. These results demonstrate that the proposed curriculum further refines class-level recognition even for a backbone that already achieves high baseline performance.

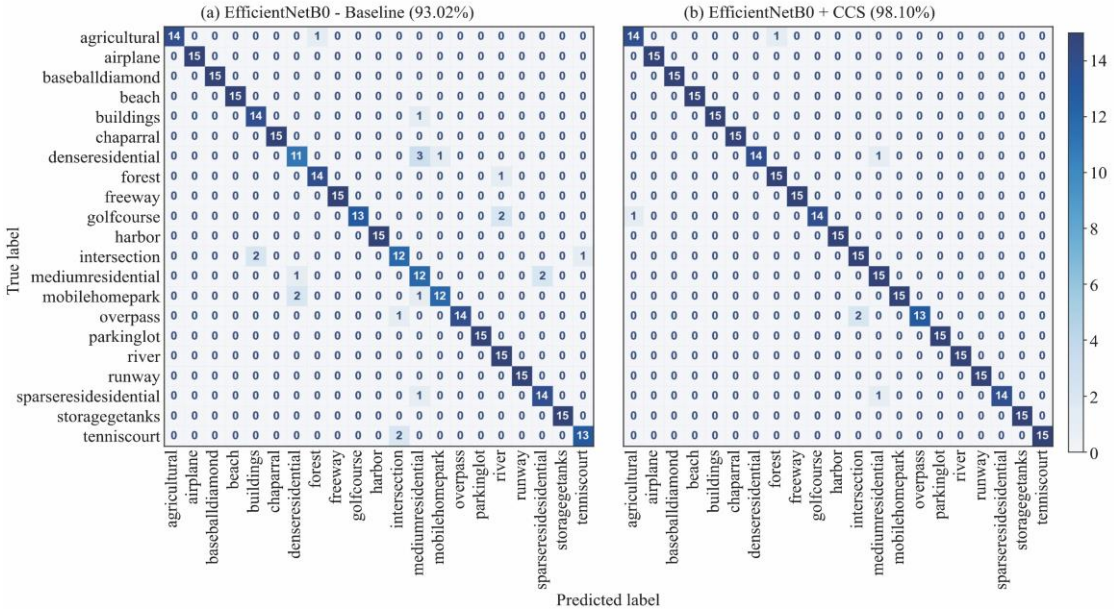

Fig. 14 Comparison of the confusion matrices obtained by (a) The baseline EfficientNet-B0, and (b) The proposed CCS-guided curriculum model on the held-out original-image test set.

Figure 15 compares the confusion matrices of the baseline MobileNetV2 model and the proposed CCS-guided curriculum. Compared with the baseline, the curriculum improves or preserves recall for 20 of the 21 scene classes, demonstrating that the proposed framework consistently enhances class-level recognition while maintaining the already strong baseline performance.

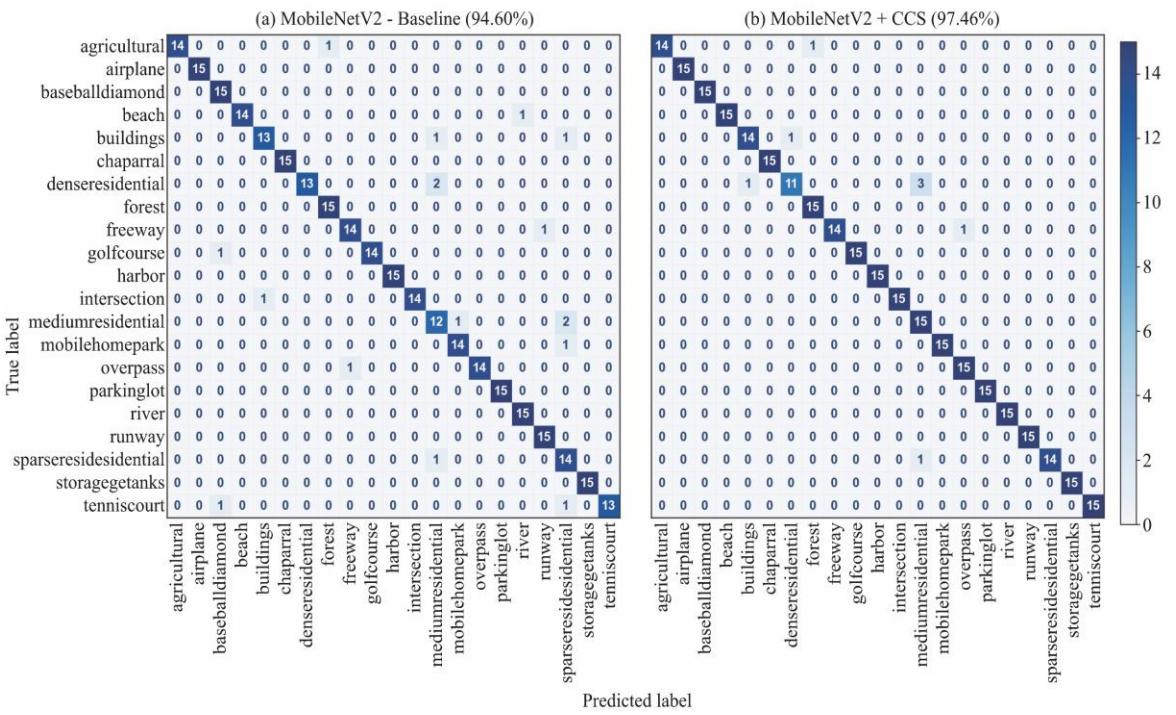

Fig. 15 The baseline MobileNetV2 and the proposed CCS-guided curriculum model were tested on the held-out original-image test set, and the confusion matrices are shown

The largest improvement is observed for medium residential, where recall increases from 80.0% to 100.0% (+20.0 pp), followed by tennis court (86.7% to 100.0%, +13.3 pp). Additional improvements of 6.7 pp are obtained for beach, buildings, golf course, intersection, mobile home park, and overpass. The only notable reduction occurs for dense residential, where recall decreases from 86.7% to 73.3% (-13.3 pp). The confusion matrices further demonstrate a stronger concentration of predictions along the principal diagonal, indicating reduced inter-class confusion for most scene categories. These findings suggest that the proposed curriculum primarily benefits visually challenging scene classes while largely preserving the high baseline performance achieved by MobileNetV2.

The class-level analysis shows that the enhancements brought by the proposed CCS-guided curriculum are not limited to a few of the scene types in the backbone architectures but are widespread across them. Especially, the visual classes with great difficulty (e.g., classes that are more vulnerable to inter-class confusion) show the greatest improvement in a consistent manner, whereas the recognition performance of the easy classes is largely preserved. The improvements to the dataset showed that the proposed curriculum improves classification accuracy not only overall, but also at a class level.

5.4. CCS Component Weight Selection

The proposed CCS framework captures the difficulty of the class by combining four complementary factors, including similarity, confusion, uncertainty, and variance. The final CCS formulation assigns weights of 0.60, 0.05, 0.25, and 0.10 to these components, respectively, based on the grid-search evaluation. This weighting scheme was used for all subsequent experiments due to the fact that it resulted in an overall improved classification performance on the held-out original-image test set while also giving a uniform curriculum progression.

As presented in Table 11, the evaluated weight configurations yielded significant differences in classification accuracy, validating the proposed curriculum's impact and the contribution of each of the CCS components. The candidate with the highest original-test accuracy (93.33%) was the one with the selected weighting, which is used as the final CCS formulation.

The results suggest that a good balance between semantic similarity, prediction confusion, predictive uncertainty, and feature variance is indispensable for the correct estimation of class difficulties and the building of an effective curriculum. Further investigation of the contribution of each CCS component is conducted.

Table 11. Weight configurations of candidates of the CCS applied to the held-out original test set

Weights (Ws, Wc, Wu, Wv)	Accuracy	Precision	Recall	F1	Cohen's κ	mAP
0.60, 0.05, 0.25, 0.10	**93.33%**	**93.60%**	**93.33%**	**93.31%**	**0.930**	**97.02%**
0.30, 0.05, 0.55, 0.10	93.02%	93.09%	93.02%	92.88%	0.9267	96.83%
0.70, 0.05, 0.20, 0.05	92.06%	92.34%	92.06%	92.02%	0.9167	96.32%
0.25, 0.25, 0.25, 0.25	91.75%	92.09%	91.75%	91.64%	0.9133	96.14%
0.45, 0.20, 0.30, 0.05	91.75%	92.23%	91.75%	91.64%	0.9133	96.56%
0.60, 0.10, 0.25, 0.05	91.43%	91.98%	91.43%	91.36%	0.9100	96.32%
0.65, 0.10, 0.20, 0.05	91.43%	92.11%	91.43%	91.39%	0.9100	96.33%
0.50, 0.10, 0.30, 0.10	91.11%	91.56%	91.11%	91.12%	0.9067	96.57%
0.55, 0.10, 0.30, 0.05	90.79%	91.20%	90.79%	90.71%	0.9033	96.35%
0.50, 0.15, 0.30, 0.05	90.79%	91.32%	90.79%	90.73%	0.9033	96.65%
0.55, 0.05, 0.35, 0.05	90.48%	91.32%	90.48%	90.43%	0.9000	96.57%
0.50, 0.10, 0.35, 0.05	90.48%	90.85%	90.48%	90.36%	0.9000	96.47%
0.45, 0.10, 0.40, 0.05	90.48%	90.86%	90.48%	90.36%	0.9000	96.77%

5.5. Ablation Study: Contribution of Individual CCS Components

An ablation study was conducted to quantify the impact of each component of the proposed CCS by removing one component at a time while maintaining the same three components and training settings. Four ablation variants are evaluated with the complete CCS framework used as a reference model to examine the role of each component in curriculum building and overall classification performance. As illustrated in Table 12, there are specific ways in which each CCS component affects the curriculum structure.

Table 12. Scene class distribution for all stages of the curriculum for the full CCS and the ablation variants

Variant	Stage 1	Stage 2	Stage 3	Stage 4
Full CCS	**5**	**10**	**17**	**21**
No Similarity	20	21	21	21
No Confusion	5	10	18	21
No Uncertainty	6	12	21	21
No Variance	6	10	19	21

Without the semantic similarity component, the curriculum flattens to the plan; 20 of the 21 classes are added to the Easy level, and all classes are presented at Stage 2, negating the progressive learning approach. Eliminating the prediction confusion component results in a minor change, and an extra class moves from Stage 2 to Stage 3 (18 cumulative classes versus 17). Excluding predictive uncertainty has a more prominent impact by removing the final stage of the curriculum altogether, with 12 cumulative classes in Stage 2 and 21 classes in Stage 3. Likewise, the 3 and 4 single-class differences in the four stages, respectively, are 6 and 10, respectively, after the feature variance component is removed, suggesting moderate redistributions of class difficulty. The observations show that semantic similarity is the main factor in the construction of the curriculum, and the other three factors provide finer detail to make the scene classes more specific in the learning stages.

As shown in Table 13, the classification performance is consistent with the changes in the curriculum in Table 12. The similarity component contributed the most to the fall in performance, as its removal led to the largest decrease in accuracy from 93.33% to 88.57% (-4.76 pp), thus showing its strong influence in building an effective curriculum. Successively discarding the confusion component decreased the accuracy to 90.79% (-2.54 pp), showing that the confusion component brings additional information beyond the semantic similarity. The uncertainty item resulted in an accuracy of 91.11% (-2.22 pp), showing that uncertainty contributes to the accurate estimation of the difficulty of classes. With the removal of the variance component, the minimum reduction in accuracy was found to be 92.06% (-1.27 pp), but the complete CCS framework still gave the maximum performance. Overall, the progressive reduction in performance across all four ablation variants demonstrates that each CCS component contributes to the effectiveness of the proposed curriculum, with similarity providing the greatest individual contribution while the remaining components offer complementary benefits.

Table 13. Performance comparison of the complete CCS framework and the single-component ablation variants on the held-out original test set

Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Cohen's κ	mAP (%)
Full CCS	**93.33**	**93.60**	**93.33**	**93.31**	**0.930**	**97.02**
No Similarity	88.57	89.04	88.57	88.47	0.880	93.96
No Confusion	90.79	90.95	90.79	90.72	0.903	96.58
No Uncertainty	91.11	91.37	91.11	91.05	0.907	96.37
No Variance	92.06	92.33	92.06	91.98	0.917	97.46

5.6. Statistical Validation of Curriculum Effectiveness

To verify that the performance improvements are genuine rather than incidental variation, the pre-specified statistical validation was applied identically to all three backbone architectures. Each backbone's best baseline checkpoint was compared against its corresponding full CCS-guided curriculum checkpoint (last stage) using the same held-out original-image test set. The validation framework evaluates five complementary aspects of the proposed method, and overall validation results are summarized in Table 14.

Table 14. Summary of statistical validation results for the proposed CCS-guided curriculum

Validation Objective	Statistical Test	Overall Result	Practical Interpretation
H1. Curriculum effectiveness	One-sided Wilcoxon signed-rank test	Significant for all three backbones	The proposed curriculum significantly improves class-level classification performance.
H2. CCS difficulty validity	Spearman rank correlation	Positive significant correlation	Higher CCS scores consistently correspond to scene classes that are intrinsically more difficult to classify.
H3. Similarity confusion mechanism	Spearman rank correlation	Positive significant correlation	Visually similar scene classes exhibit greater inter-class confusion, validating the semantic similarity component of CCS.
Image-level improvement	Directional exact McNemar test	Significant for all three backbones	The curriculum correctly classifies significantly more test images than it misclassifies relative to the baseline models.
Robustness of improvements	Paired bootstrap (5,000 resamples)	Supported for all major metrics	Performance improvements remain stable under repeated resampling, with only the MobileNetV2 mAP improvement not reaching statistical significance.

H1 (Curriculum effectiveness): To determine whether the proposed curriculum consistently improves class-level recognition performance, a one-sided Wilcoxon signed-rank test was performed on paired per-class F1-scores. After Holm–Bonferroni correction, statistically significant improvements were observed for all three backbone architectures: Custom CNN (median $\Delta F1 = +0.028$, Holm-adjusted $p = 0.0019$), EfficientNet-B0 (median $\Delta F1 = 0.000$, Holm-adjusted $p = 0.0068$), and MobileNetV2 (median $\Delta F1 = +0.032$, Holm-adjusted $p = 0.0085$). The zero median improvement for EfficientNet-B0 is due to the fact that many of the scene classes in it had already reached the ceiling level in recognition. However, the test does remain statistically significant, as the remaining challenging classes are still enjoying the benefits of the proposed curriculum with the positive mean $\Delta F1 = +0.050$.

H2 (CCS difficulty validity): Spearman's rank correlation was computed between each class's original CCS value and its baseline classification difficulty (1 − baseline F1-score) to check whether the proposed CCS is able to accurately estimate the intrinsic class difficulty. Positive and statistically significant correlations were obtained for all three backbone architectures: Custom CNN ($\rho = 0.628$, Holm-adjusted $p = 0.0023$), EfficientNet-B0 ($\rho = 0.605$, Holm-adjusted $p = 0.0068$), and MobileNetV2 ($\rho = 0.597$, Holm-adjusted $p = 0.0085$). These results demonstrate that scene classes assigned higher CCS values are consistently more difficult for the baseline models to classify, thereby validating the proposed difficulty estimation strategy.

H3 (Similarity–confusion mechanism validity): Since semantic similarity constitutes the largest component of the proposed CCS formulation, an additional hypothesis was evaluated to determine whether visually similar scene categories are genuinely more susceptible to inter-class confusion. Spearman correlation between pairwise feature similarity and pairwise baseline confusion was positive and highly significant for all three backbone architectures: Custom CNN ($\rho = 0.274$, Holm-adjusted $p = 1.7 \times 10^{-4}$), EfficientNet-B0 ($\rho = 0.331$, Holm-adjusted $p = 2.7 \times 10^{-6}$), and MobileNetV2 ($\rho = 0.331$, Holm-adjusted $p = 2.7 \times 10^{-6}$). These findings provide direct empirical support for incorporating semantic similarity into the CCS formulation and explain why similarity contributes most strongly to curriculum construction, as demonstrated in the ablation study.

Supporting analyses. The effectiveness of the proposed curriculum has also been seen to be effective through image-level statistical validation. A directional exact McNemar test was conducted, demonstrating that for all three backbone architectures, including Custom CNN, EfficientNet-B0, and MobileNetV2, the CCS-guided curriculum corrected significantly more test images than harmed. As a result, improvement reported in any part of this study is real improvement at the sample level, and not just a random fluctuation in prediction. Then, paired bootstrap analysis with 5,000 resamples was conducted and showed that the improvements in Accuracy, macro-F1 score, and Cohen's Kappa for the three backbone architectures were robust. The MobileNetV2 mAP improvement was the only result that did

not reach statistical significance ($\Delta = 0.005$, 95% CI = [-0.005, 0.017]), but for the other neural networks in the statistical validation framework, such support was obtained for mAP. Overall, the statistical validation supports the proposed CCS-guided curriculum from various complementary perspectives consistently.

The Wilcoxon test shows that there are significant improvements in the class-level performance; Spearman correlation is used to validate the proposed difficulty estimation strategy and the similarity-based learning mechanism, and the McNemar test confirms the consistency of the image-level improvements; and the results of the bootstrap analysis show the robustness of the observed performance gains. All of these stand-alone statistical analyses confirm that the gains made by the proposed CCS-driven curriculum are statistically proven, theoretically proven, and reproducible with different backbone architectures.

5.7. Discussion

Under a common experimental protocol, the CCS-guided curriculum resulted in statistically significant improvements in performance across all three backbone architectures. The custom CNN did best, perhaps because the feature learning process is directly affected by the curriculum progression. The framework is also compatible with randomly initialized architectures, as well as pretrained architectures, such as EfficientNet-B0 and MobileNetV2. The ablation results show how complementary the similarity, confusion, uncertainty, and variance are. The other factors only help to further differentiate levels of difficulty in the curriculum, with similarity being the main factor that sets up the curriculum structure. Their individual contributions show that the four signals have different aspects of class complexity.

Wilcoxon signed-rank tests, McNemar's test, and bootstrapping tests also support the consistent improvement across scene classes and test the reliability of performance. CCS values and baseline classification difficulty support the use of CCS as an indicator of class-level difficulty in learning.

Images were enhanced during training and only selected challenging classes, and original images were used for validation and testing. This way, the performance improvements were measured on the same image distribution and not an enhancement-matched domain. Similar experimental setups within the baseline and curriculum models also enhance the comparability of the two models. CCS offers a data-based method for difficulty level determination that is complementary to the traditional method of manually creating levels of difficulty, by incorporating four distinct signals of difficulty at the class level during curriculum construction. The outcomes show effectiveness on the UC Merced data set, but more testing needs to be done to prove the generalizability of the results on other remote sensing data sets and classification tasks.

The results of the present work could also be related to the various existing methods of curriculum learning and remote sensing scene classification. Various strategies to structure the learning process have been discussed in previous studies of easy-to-hard multi-class learning [21], curriculum-assisted hierarchical vision transformers [22], and curriculum-based incremental learning [23]. The proposed CCS-guided framework, however, combines inter-class similarity, prediction confusion, model uncertainty, and intra-class variance into a class-level difficulty assessment and provides a framework for backbone-specific learning sequences, progressive class introduction, and class-specific enhancement. This methodological difference is corroborated by the regular increase in performance for the three evaluated architectures. In particular, the accuracy rose by 5.40%, 5.08%, and 2.86% for the custom CNN, EfficientNetB0, and MobileNetV2, respectively. The findings indicated that learning by class complexity can enhance the capability of discriminating features by gradually presenting more and more difficult and similar classes.

Moreover, the proposed framework focuses on optimizing the training strategy by varying class progression based on the training difficulty and adaptively improving the training difficulty, unlike the data-correction and lightweight ensembling method explored by Song et al. [8]. The effectiveness of the proposed components is also confirmed by the ablation and statistical analyses. The improvements over the conventional training protocols are consistent across lightweight architectures, proving the effectiveness of CCS-guided curriculum learning across different architectures.

6. Challenges and Future Directions

The following section outlines the key methodological and experimental challenges identified in the present study, followed by potential future research directions to address these limitations and further improve the robustness, adaptability, and generalizability of the proposed CCS-guided curriculum learning framework.

6.1. Challenges

This section presents methodological and experimental limitations found in the current study and potential directions for future research to overcome these limitations and make the proposed CCS-guided curriculum learning framework more robust, adaptable, and generalizable.

- Limited experimental generalizability: The evaluation was performed on the UC Merced dataset, one data split (70/15/15), and one training run per configuration. The small number of test samples may impact the stability of the performance estimates, and applicability to larger datasets, different architectures, and other types of remote sensing tasks has not been demonstrated.
- Experimental comparability: Differences in the amount of augmentation, amount of training data, and length of training between baseline and curriculum settings make it

hard to be certain about the contribution of curriculum learning from any additional exposure to data and optimization.

- The sensitivity and uncertainty of the CCS: Class rankings may vary during the training process and across datasets, due to fixed component weights, a pre-defined difficulty threshold, and one-time estimation of CCS. In addition, using softmax-based confidence is not sufficient to represent predictive uncertainty, especially in cases of poor calibration of the model's predictions.
- Enhancement Limitations: The six manually designed RGB enhancement techniques are based on using selected spectral characteristics, but they do not use multispectral information. Their effectiveness could thus vary with the characteristics of the data sets and may not be consistently transferred to different imaging data sets.

Limited architectural and task coverage: Three convolutional backbones were used for the evaluation of the framework in the context of single-label scene classification. These are not things that have been explored with it, including transformer-based architectures, larger models, and dense prediction tasks.

6.2. Future Directions

- Wide-ranging validation: Future work should involve other random seeds, cross-validation, and larger remote sensing datasets, such as AID, NWPU-RESISC45, and EuroSAT, for validation and robustness and generalizability tests.
- Precise component evaluation: Putting together a standard augmentation, a fixed training set size, and a fixed computational budget for all configurations, as well as having a curriculum-only and enhancement-only experiment, would help isolate the contribution of each component.
- Adaptive CCS formulation: Future research could involve varying the difficulty of each component of the CCS, uncertainty estimation using a data-driven approach, and adaptive updating of the CCS throughout the curriculum to enhance the stability and responsiveness of the ordering of the classes.
- Advanced Enhancement Strategy: Where appropriate spectral bands are available, selected Multispectral Indices (MSI) such as NDVI and NDWI might be investigated to assess how well they may improve the adaptability and effectiveness of class-specific enhancement.
- Expansion to other architectures and tasks: Testing the framework using vision transformers and extending to semantic segmentation and other classification scenarios in remote sensing could show its general applicability.

7. Conclusion

In this study, a framework of CCS-guided curriculum learning for remote sensing land-use scene classification was introduced. The framework ranks the classes by four complementary difficulty indicators, enhances difficult classes with RGB enhancement at the training stage, and progressively improves the models in four incremental curriculum stages. The framework was evaluated on the UC Merced dataset with three backbone architectures with classification accuracy of 93.33%, 98.10%, and 97.46% for the custom CNN, EfficientNetB0, and MobileNetV2, respectively. The custom CNN trained from scratch gained the highest amount of improvement (+5.40 pp) using 1.74 million parameters. The gains were confirmed with statistical analysis and complemented with the results of the ablation experiments, which showed the importance of inter-class feature similarity in the curriculum structure, and the ambiguity, uncertainty, and variance as complementary information for difficulty. The results show that difficulty-aware training can enhance the performance of classification without changing the backbone architectures, thereby proving the effectiveness of the proposed method. There are still a number of restrictions, such as using one set of data, one data split, and one training run for each configuration. The individual contributions are also limited because there are differences between the baseline and curriculum models in terms of the augmentation, they are a part of training-set size, and cumulative training exposure. Moreover, CCS is based on static component weights and difficulty levels, and ablation analysis was only conducted on the custom CNN. Future work will include studying robustness for various random seed sets and scaling up to larger benchmarks, baseline comparisons with controlled difficulty, dynamic difficulty estimation, further enhancement methods, and extending to transformer-based and semantic segmentation architectures. The introduction should be succinct, with no subheadings. Limited figures may be included only if they are truly introductory and contain no new results.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

Not Applicable.

List of Abbreviation

Abbreviation	Full Form
CCS	Class Complexity Score
RGB	Red, Green, Blue
UC Merced	University of California, Merced
UCM	University of California, Merced
CNN	Convolutional Neural Network
EfficientNet-B0	EfficientNet, Baseline B0 Variant
MobileNetV2	MobileNet Version 2
RS	Remote Sensing

MODIS	Moderate Resolution Imaging Spectroradiometer
UN SDGs	United Nations Sustainable Development Goals
LULC	Land Use and Land Cover
AI	Artificial Intelligence
AID	Aerial Image Dataset
NWPU-RESISC45	Northwestern Polytechnical University Remote Sensing Image Scene Classification Dataset (45 Classes)
EuroSAT	European Space Agency Sentinel-2 Satellite Image Dataset
SIRI-WHU	Satellite Image Remote Sensing Image Dataset - Wuhan University
M	Million
pp	Percentage Points
VGG19	Visual Geometry Group, 19-Layer Network
GoogLeNet	Google's Inception Neural Network Architecture
DenseNet121	Densely Connected Convolutional Network, 121-Layer Variant
ResNet	Residual Network
Xception	Extreme Inception
MS-DARNet	Model-Specific Acronym (Original Publication Required)
AlexNet	Alex Krizhevsky's Convolutional Neural Network
CILEA-NET	Model-Specific Acronym (Original Publication Required)
YOLO	You Only Look Once
CLAHE	Contrast Limited Adaptive Histogram Equalization
RSSCN7	Remote Sensing Scene Classification Dataset with Seven Classes
WHU-RS19	Wuhan University Remote Sensing Dataset with 19 Classes
S	Similarity
C	Confusion

U	Uncertainty
V	Variance / Intra-Class Feature Dispersion
w	Weight
ExG	Excess Green Index
NDWI	Normalized Difference Water Index
NDVI	Normalized Difference Vegetation Index
MSI	Multispectral Imagery / Multispectral Imaging
HSV	Hue, Saturation, Value
NVIDIA	NVIDIA (Company Name)
GeForce	NVIDIA Graphics Processor Brand
RTX 3070	NVIDIA GeForce RTX 3070 GPU Model
Ti	Titanium
GPU	Graphics Processing Unit
Conv2D	Two-Dimensional Convolution
BatchNorm	Batch Normalization
ReLU	Rectified Linear Unit
MaxPool	Max Pooling
Softmax	Softmax Activation Function
MBConv	Mobile Inverted Bottleneck Convolution
stem_conv	Stem Convolution Layer
SiLU	Sigmoid Linear Unit
top_conv	Top Convolution Layer
2D	Two-Dimensional
Adam	Adaptive Moment Estimation
F1-score	F1 Measure
mAP	Mean Average Precision
H1	Hypothesis 1
H2	Hypothesis 2
H3	Hypothesis 3
%	Percentage
ΔF1	Change in F1-score
p	p-value
Holm-adjusted	Holm-Bonferroni Multiple-Comparison Adjustment
CI	Confidence Interval

References

[1] Alexandra Bell et al., "Scientific Evidence from Space-A Review of Spaceborne Remote Sensing Applications at the Science–Policy Interface," *Remote Sensing*, vol. 15, no. 4, pp. 1-32, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Qiang Zhao, and Le Yu, "Advancing Sustainable Development Goals through Earth Observation Satellite Data: Current Insights and Future Directions," *Journal of Remote Sensing*, vol. 5, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Hao Wu et al., "A Data- and Expert-Driven Framework for Establishing Land Cover–Related Essential Variables for SDG Monitoring and Assessment," *Geography and Sustainability*, vol. 7, no. 1, pp. 1-11, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[4] Gong Cheng et al., "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735-3756, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Aakash Thapa et al., "Deep Learning for Remote Sensing Image Scene Classification: A Review and Meta-Analysis," *Remote Sensing*, vol. 15, no. 19, pp. 1-37, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Hussain Mobarak Albarakati et al., "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*, vol. 17, pp. 6338-6353, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Aisha M. Mashraqi et al., "Hybrid Deep Learning and Optimization-Based Land Use and Land Cover Classification for Advancing Sustainable Agriculture in Najran City, Saudi Arabia," *Scientific Reports*, vol. 15, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Huaxiang Song et al., "Pure Data Correction Enhancing Remote Sensing Image Classification with a Lightweight Ensemble Model," *Scientific Reports*, vol. 15, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jiawei Huang, and Chengjun Xu, "MS-DARNet: A Lightweight Multi-Scale Selective Dilated Attention Residual Network for Remote Sensing Scene Classification," *Remote Sensing*, vol. 18, no. 8, pp. 1-45, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[10] Xuefei Xu, and Chengjun Xu, "Fusing Geometric and Semantic Features via Cosine Similarity Cross-Attention for Remote Sensing Scene Classification," *Sensors*, vol. 26, no. 5, pp. 1-31, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yang Chu, Minchao Ye, and Yuntao Qian, "Fine-Grained Image Recognition Methods and their Applications in Remote Sensing Images: A Review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 19640-19667, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Petru Soviany et al., "Curriculum Learning: A Survey," *International Journal of Computer Vision*, vol. 130, no. 6, pp. 1526-1565, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Yi Yang, and Shawn Newsam, "Bag-of-Visual-Words and Spatial Extensions for Land-Use Classification," *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 270-279, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fan Hu et al., "Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14680-14707, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[15] Tsion Fekadu Deressu et al., "Enhancing Land Use and Land Cover Classification with Deep Learning-Based Satellite Imagery Segmentation," *International Journal of Applied Earth Observation and Geoinformation*, vol. 144, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed, "A Review of Deep Transfer Learning and Recent Advancements," *Technologies*, vol. 11, no. 2, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Chu He et al., "Relationship Prior and Adaptive Knowledge Mimic Based Compressed Deep Network for Aerial Scene Classification," *IEEE Access*, vol. 7, pp. 137080-137089, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Russo Ashraf, and Kang-Hyun Jo, "Frequency and Texture Aware Multi-Domain Feature Fusion for Remote Sensing Scene Classification," *IEEE Access*, vol. 13, pp. 16380-16393, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nan Mo, and Ruixi Zhu, "Semi-Supervised Subcategory Centroid Alignment-based Scene Classification for High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 16, no. 19, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Nourhan H. Farag et al., "Enhancing the Classification of Spectrally Similar Land Use/Land Cover Classes using Transfer Learning in Arid Regions," *Scientific Reports*, vol. 16, pp. 1-12, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[21] Weiwei Liu, Ivor W. Tsang, and Klaus-Robert Müller, "An Easy-to-Hard Learning Paradigm for Multiple Classes and Multiple Labels," *Journal of Machine Learning Research*, vol. 18, no. 94, pp. 1-38, 2017. [Google Scholar] [Publisher Link]

[22] Yi Liu et al., "A Hierarchical ViT With Dynamic Window Shift Unit and Curriculum Learning for Remote Sensing Image Scene Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 8011-8024, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] S. Divakar Bhat et al., "CILEA-NET: Curriculum-Based Incremental Learning Framework for Remote Sensing Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 5879-5890, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Wandi Cai, Bo Zhang, and Bin Wang, "Scale-Aware Anchor-Free Object Detection via Curriculum Learning for Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9946-9958, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Peizhuo Liu et al., "A Semi-Supervised Transformer with a Curriculum Training Pipeline for Remote Sensing Image Semantic Segmentation," *Remote Sensing*, vol. 18, no. 3, pp. 1-34, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[26] N. Shobha Rani et al., "A Novel Curriculum Learning Training Strategy for Pomegranate Growth Stage Classification Using YOLO Models on Multi-Source Datasets for Precision Agriculture," *IEEE Access*, vol. 13, pp. 112594-112622, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[27] Sheng-Chieh Hung, Hui-Ching Wu, and Ming-Hseng Tseng, "Integrating Image Quality Enhancement Methods and Deep Learning Techniques for Remote Sensing Scene Classification," *Applied Sciences*, vol. 11, no. 24, pp. 1-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] László Bakacsy et al., "Drone-based Identification and Monitoring of Two Invasive Alien Plant Species in Open Sand Grasslands by Six RGB Vegetation Indices," *Drones*, vol. 7, no. 3, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[29] Lavinia Moscovini et al., "Vegetation Indices Prediction Using Machine-Learning and Deep-Learning Approaches From RGB Calibrated Images," *IEEE Access*, vol. 13, pp. 203216-203229, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[30] Xuemei Li, Huping Ye, and Shi Qiu, "Cloud Contaminated Multispectral Remote Sensing Image Enhancement Algorithm based on MobileNet," *Remote Sensing*, vol. 14, no. 19, pp. 1-27, 2022. [CrossRef] [Google Scholar] [Publisher Link]