Zahra: Bulletin Big data, Data Science and Artificial Intelligence
Vol. 2, No. 2, [December 2023], pp. [87–97]
E-ISSN 2830-7984

Research Article Open Access (CC–BY-SA)

Optimizing Distance-Weighted Manhattan K-Nearest Neighbors with Spectral Standardization for Landsat Land-Cover Classification

Hartini Damanik[1], Lidya Wati[2*], Abdul Karim[3], Hilmi Aziz Bukhori[4]

[1] *Informatics, STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia.*
[2] *Faculty of Intelligent Computing, Universiti Malaysia Perlis (UniMAP), Malaysia*
[3] *Information Technology Study Program, Faculty of Science and Technology, Universitas Labuhanbatu, Indonesia.*
[4] *Mathematics Department, Faculty of Mathematics and Natural Sciences, Universitas Brawijaya, Indonesia*

[1] *hartinidamanik12@gmail.com;* [2] *lidyawati@studentmail.unimap.edu.my;* [3] *Abdulkarim@ulb.ac.id;* [4] *hilmibukhori@ub.ac.id*
** Corresponding author*

Abstract

Distance-based Landsat land-cover classification depends on feature scaling, neighborhood geometry and voting rules. This study compares two standardized K-nearest-neighbor classifiers on the official UCI Statlog Landsat Satellite split of 4,435 training and 2,000 test records. Both models retain all 36 spectral attributes and fit StandardScaler only on the respective training data. The baseline uses Euclidean distance, uniform voting and five neighbors. The proposed configuration fixes Manhattan distance and inverse-distance voting while selecting k from 1 to 35 through eight training-only candidate evaluations. The selected k is three, with internal validation macro-F1 of 0.886329. The configured TPESampler remains within its documented random startup phase at this budget. On the official test set, the baseline achieves 90.45% accuracy and 89.16% macro-F1; the selected model achieves 90.85% and 89.64%. Using unrounded scores, these correspond to gains of 0.40 and 0.49 percentage points. Class-5 recall rises from 87.34% to 91.56%, whereas class-4 recall falls from 68.72% to 67.77%. The comparison therefore shows a small aggregate improvement with uneven class effects. Because distance, voting and k change together, the experiment does not isolate their individual contributions. A single seed, a limited search budget and the absence of geographic identifiers restrict the conclusion to this fixed benchmark; broader optimizer, component and transferability claims require further evaluation.

Keywords: Landsat land-cover classification; K-nearest neighbors; Manhattan distance; inverse-distance voting; spectral standardization

1. Introduction

The growing demand for defensible land-cover information has made classification of multispectral Landsat measurements an important research problem. Reviews of regional mapping products and deep-learning-based land-cover classification show that reported performance depends on representation, training observations and evaluation design [1], [2], [3], [4]. Feature-relationship models and vegetation-index or texture-based studies further illustrate the importance of how spectral information is represented [5], [6], [7]. These advances do not make a compact distance-based baseline irrelevant: methodological surveys describe nearest-neighbor classification as an established alternative whose behavior depends on the available examples and the definition of similarity [8], [9]. In such a model, standardization can change the relative influence of input attributes, while the distance measure and neighbor weights alter how local evidence is combined [10], [11]. This makes standardized KNN a useful setting for an explicit comparison of neighborhood rules rather than a claim that a new classifier family has been invented. The present experiment retains the 36 original Statlog attributes in both arms and uses the same official training and test records. Its research question is whether a training-selected distance-weighted Manhattan configuration changes aggregate and class-wise performance relative to a standardized Euclidean, uniform-vote baseline.

The methodological challenge is to separate the construction of a candidate configuration from its final assessment. Hyperparameter-search studies emphasize the role of the objective, bounds, initialization and computational budget [12], [13], [14]. A search can favor a particular validation subset, so the score used to select a configuration is not interchangeable with an independently reserved test score [15], [16], [17]. Benchmarking and reproducibility research also requires a specific comparator and an explicit data boundary before interpreting a small difference [18], [19], [20]. These concerns are material for archived spectral patches: spatial or other dependencies can make a random partition less demanding than a test on separate geographic scenes [21], [22], [23], [24]. In addition, aggregate accuracy can conceal changes in smaller or spectrally ambiguous classes; macro averages, per-class measures and confusion matrices provide complementary information [25], [26], [27], [28]. Calibration and operational costs require separate evidence and should not be inferred from class labels or accuracy alone [29], [30]. The comparison here therefore reports the endpoints actually available from the notebook rather than unmeasured confidence quality, query latency or deployment robustness.

To address this question, the study evaluates two StandardScaler–KNN pipelines under the official UCI Statlog partition. The fixed baseline uses five neighbors, Euclidean distance and uniform votes. The selected alternative uses Manhattan distance, inverse-distance votes and a training-selected neighbor count. Only k is searched; the input representation and the standardization procedure are shared. Configuration selection uses a stratified 75:25 split of the official training records, followed by refitting on the complete official training set and evaluation on the unchanged test set. This design follows the general principle of separating selection from final assessment [16], [19], [31] . Separate unscaled, Manhattan-uniform and exponent-tuned models were not executed, so the comparison does not provide a four-component ablation. The contribution is consequently a traceable comparison of two neighborhood configurations, with class-level errors and internal learning curves, rather than a verified interaction among independent innovations. Learning-curve and validation literature motivates treating the internal curves as diagnostics within the development population [31], [32]. The conclusions remain limited to one split and seed, and do not establish spatial transferability or a statistically stable optimizer advantage.

2. Method

2.1. Related Work

Recent land-cover reviews connect classifier performance with the source observations, training-data design and the spatial or thematic scope of the resulting map [1], [2], [3], [4] . Landsat feature-relationship research provides a direct example of representation-driven classification [5]. These studies establish the broader remote-sensing context, but they do not validate the particular scores obtained from Statlog or establish that one neighbor rule is generally preferable. The present benchmark is a classification of archived multispectral patch records, rather than a newly produced geographic map.

Engineered vegetation indices, texture features and learned spectral–spatial representations have been explored to expose class distinctions [6], [7]. Multisensor optical–radar fusion offers a further example of combining complementary observations [33]. General feature-engineering and selection reviews discuss transformations, redundant predictors and the role of a declared input representation [34], [35]. These alternatives motivate stating exactly which measurements enter a classifier. Here, both KNN arms use the same 36 original attributes; no new spectral indices, feature selection or additional sensor data are introduced.

Nearest-neighbor methods involve both a classification rule and the computational task of finding nearby observations. A classical-classifier survey describes their methodological characteristics [8], while a review of exact kNN search and join methods focuses on efficient neighbor retrieval and high-dimensional data [9]. A more directly relevant classification study evaluates combinations of nearest-neighbor weighting, distance and scaling choices [11]. Its alternatives include weighting schemes beyond the inverse-distance rule used here. These papers support examining neighborhood design explicitly, rather than treating all KNN variants as equivalent or assuming that a more complex voting rule always improves a particular dataset.

Scaling is especially relevant when similarity is calculated from several numerical attributes. Comparative scaling research shows that preprocessing choices can materially affect classification outcomes [10]. Feature-engineering guidance likewise emphasizes making transformations and their fit boundaries explicit [35]. Leakage research explains why statistics estimated from evaluation observations should not enter a training pipeline [15], [16]. The present experiment applies the same training-fitted StandardScaler procedure to both models. This controls the scaling choice in the model comparison; it does not establish an empirical benefit of standardization over an unscaled baseline.

Hyperparameter-optimization and automated-learning reviews describe search as a procedure whose outcome depends on the parameter space, objective, validation and budget [13], [14]. Satellite-imagery classification studies have examined how configuration choices affect performance [12], and recent explainable multisensor land-cover work combines model selection with a richer remote-sensing setting [36]. These studies motivate reporting the actual search domain and selected value. The current notebook searches the integer neighbor count from 1 to 35 while keeping Manhattan distance and inverse-distance voting fixed; it does not jointly optimize an additional distance exponent.

Some optimization studies consider multiple objectives, including fairness, predictive error and energy use [37], [38]. Deployment research additionally documents costs and constraints beyond predictive accuracy [29]. Such concerns are relevant to choosing a practical neighbor classifier, but they require measurements on a declared environment. The present notebook supplies classification outputs and a limited selection budget, without runtime, memory or energy benchmarks. It therefore supports a comparison of recorded classification scores rather than claims that the selected model offers a better accuracy–latency or accuracy–energy trade-off.

Multiclass metric guidance distinguishes overall accuracy, per-class precision and recall, macro averages and support-weighted summaries [25], [27]. Image-validation research identifies how metric choice and aggregation can conceal important failures [26], [28]. These considerations motivate reporting the complete classification table and both absolute and row-normalized confusion matrices. Predictive calibration is a different property from correct class assignment [30]; class-conditional conformal uncertainty mapping offers another reliability perspective in land-cover classification [39]. Neither calibration nor conformal prediction is evaluated here, so neighbor vote shares are not treated as validated confidence estimates.

Evaluation boundaries are also central to reproducible model comparison. Leakage and common-pitfall studies describe the risks of reusing evaluation information or comparing methods under inconsistent preprocessing [15], [16], [17]. Reproducibility recommendations require explicit model and data specifications [19]. Benchmarking research further shows that conclusions depend on the selected datasets and comparator design [18], [20], while work on cross-validation clarifies the target of an estimated performance measure [31]. The current fixed baseline and selected alternative therefore answer a limited configuration-comparison question; they do not provide an equal-budget search comparison across optimizer families.

Geospatial validation methods distinguish random partitions from evaluations that address geographic dependence [22], [23], [24]. Remote-sensing accuracy studies also show that an evaluation arrangement can produce overly favorable results or conceal location-dependent error [21], [40]. Cross-domain and cross-scene hyperspectral models address transfer through different architectures and data conditions [41], [42]. These contributions support a careful geographic interpretation of this benchmark: local measurements from 3×3 patches are available, but reconstructable scene identifiers for spatial blocking are not supplied to the notebook. An improvement on its official test set cannot be generalized directly to separate acquisition regions.

Learning-curve research describes how predictive behavior can change with training-set size and why curve shape depends on the learner and data [32]. Validation theory also cautions against interpreting a reused development partition as independent replication [31]. The notebook curves refit the fixed baseline and the already selected Manhattan configuration at four training sizes while retaining one internal validation subset. These diagnostics complement the official-test comparison, but do not measure repeated-seed stability or isolate the effects of distance, weighting and k. The distinction is especially relevant for distance-weighted KNN, whose resubstitution score can be dominated by zero-distance self-matches.

2.2. Dataset

The experiment uses the UCI Statlog Landsat Satellite benchmark. The notebook loads sat.trn with 4,435 observations and sat.tst with 2,000 observations. Each record contains 36 numerical spectral attributes arranged as four channels at each position of a 3×3 neighborhood, together with the central-pixel class. The six observed labels are 1, 2, 3, 4, 5 and 7. Table 1 specifies the actual input and evaluation boundary.

Table 1. Dataset and official experimental partition

Dataset attribute	Study-specific protocol
Public source	UCI Statlog Landsat Satellite: official sat.trn and sat.tst
Research endpoint	Six observed labels: 1, 2, 3, 4, 5, 7
Analysis unit	One 3×3×4 patch record, 36 attributes and its central-pixel class
Data partitions	4,435 official train / 2,000 official test; internal 3,326 / 1,109 split
Processing / quality audit	StandardScaler fitted only on the current fitting records; all 36 attributes retained
Annotation / scope check	UCI Statlog already encodes 3×3×4 neighborhoods as 36 features; spatial-geographic coordinates and a reconstructable full scene are not supplied.

As Table 1 shows, the same 36-dimensional input is retained in both arms. The feature extractor is called with mode=raw, so the helper functions for polynomial, contrast or fusion descriptors do not contribute to this experiment. The official training set is internally divided into 3,326 fitting and 1,109 validation records using stratification, test_size=0.25 and seed 42. The dataset records do not provide the geographic identifiers needed to reconstruct independent scene blocks. Table 2 records the executed development-to-test boundary. Each candidate pipeline fits its scaler and neighbor model on the 3,326-record fitting subset. After k is selected using the 1,109 validation records, a new pipeline is refitted on all 4,435 official training records. The 2,000 official test records are used only for final classification outputs.

Table 2. Experimental data flow and leakage-prevention controls

Stage	Required handling
Stage A: audit	Load the official files, check feature dimensions and display training-label counts
Stage B: split	Keep sat.trn/sat.tst separate; stratified internal 75:25 split with seed 42
Stage C: fit	For each candidate, fit StandardScaler and Manhattan KNN on the internal fitting subset
Stage D: freeze	Select k=3 from validation macro-F1, then refit the selected pipeline on all sat.trn
Stage E: report	Official-test classification metrics and confusion matrices; training-only learning curves

The controls in Table 2 keep official-test scores outside the selection objective. Macro-F1 is the selection criterion. Accuracy, weighted F1, balanced accuracy, per-class precision/recall/F1 and confusion matrices are reported on the official test set. The learning curves use only the official training records. Query latency, memory, calibration and uncertainty intervals were not measured.

2.3. Research Framework

Figure 1 presents the two executed arms under the same official data partition. Both use training-fitted StandardScaler and all 36 original attributes. Euclidean_KNN_Baseline uses uniform votes with k=5 and p=2. Weighted_Manhattan_KNN uses distance-weighted votes, p=1 and the selected k=3. The diagram identifies the shared preprocessing and the configuration differences without adding unexecuted components.

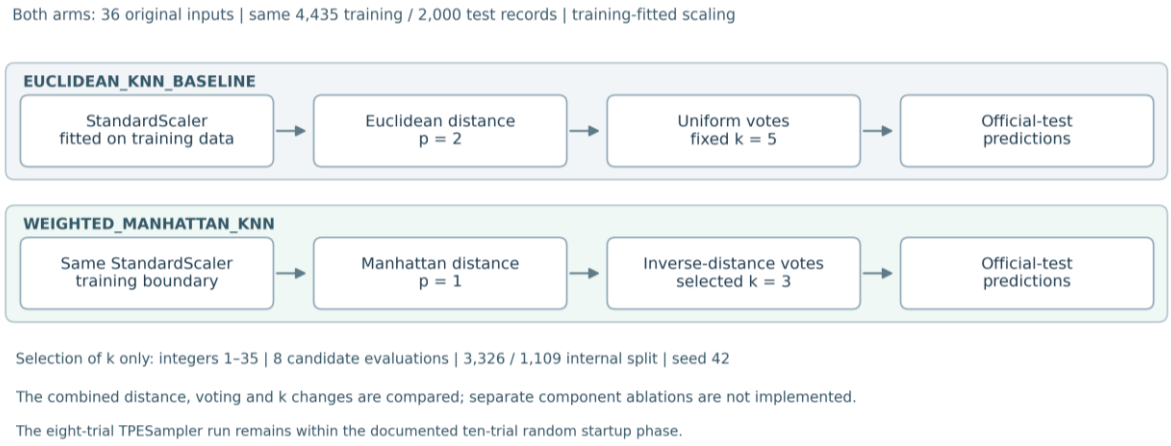


Figure 1. Controlled baseline-versus-proposed research framework

In Figure 1, standardization is a shared control, rather than an innovation evaluated against an unscaled model. Distance geometry, voting and k change together in the recorded comparison. Consequently, any observed

difference belongs to the combined configuration and cannot be assigned uniquely to one component. The fixed baseline is not separately tuned under an equal search budget. Table 3 summarizes the actual contrasts. Separate unscaled Euclidean, Manhattan-uniform, fixed-k distance-weighted and distance-exponent ablations were not executed. Their absence limits component-level interpretation of the final score difference.

Table 3. Executed standardized KNN configurations and comparison scope

Component	Comparative specification
Baseline	StandardScaler + Euclidean KNN: k=5, p=2, weights=uniform
Selected configuration	StandardScaler + Manhattan KNN: k=3, p=1, weights=distance
Controlled factors	Same 36 inputs, official split, labels, scaling procedure and evaluation code
Comparison scope	Two full configurations only; separate component and equal-budget ablations not executed
Primary comparisons	Official-test macro-F1, accuracy, weighted F1, balanced accuracy and per-class scores

Table 3 compares two complete pipelines on the same official test records. It establishes a descriptive contrast between a fixed standardized Euclidean KNN and a selected standardized weighted Manhattan KNN. It does not establish a synergistic interaction among scaling, metric and weighting or superiority over a fully tuned Euclidean comparator.

2.4. Proposed Method

The implemented method combines per-attribute standardization, Manhattan distance, inverse-distance voting and selection of the neighbor count. Figure 2 shows the executed sequence. For each fitting sample, the scaler estimates one mean and scale for each of the 36 attributes; it does not estimate only four shared band statistics. The baseline uses the same scaler procedure, with Euclidean distance and uniform voting.

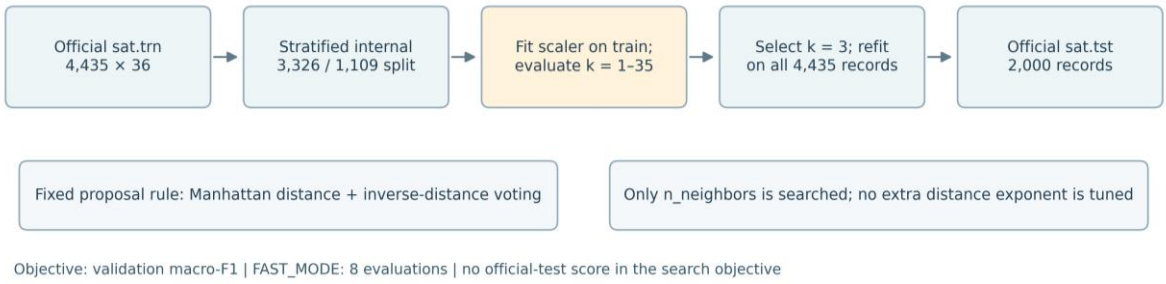


Figure 2. Training-fitted standardization and neighbor-count selection pipeline

As illustrated in Figure 2, the search evaluates integers k from 1 to 35 while p=1 and inverse-distance voting remain fixed. Each candidate pipeline contains StandardScaler followed by KNeighborsClassifier with n_neighbors=k, weights='distance', p=1 and n_jobs=1. Optuna maximizes validation macro-F1 over eight trials in FAST_MODE=True using TPESampler(seed=42). The notebook does not set n_startup_trials; under the documented default of ten, these eight evaluations remain random startup samples rather than model-guided TPE proposals. The selected k=3 achieved validation macro-F1 of 0.8863293053632062 before final refitting. Table 4 states the implemented distance and voting rule, selection objective and final configurations. These quantities follow the code and stored outputs; no adjustable voting exponent, calibrated-probability experiment or timing benchmark is inferred from the title.

Table 4. Implemented standardization, distance weighting and search settings

Design element	Operational definition
Standard input branch	Fit a mean and standard deviation for each of the 36 input attributes on fitting records
Distance and voting	Fixed Manhattan p=1 with built-in inverse-distance voting
Searched parameter	n_neighbors: integer 1–35; eight candidate evaluations; selected k=3
Output model	StandardScaler + KNeighborsClassifier(k=3, weights=distance, p=1, n_jobs=1)
Optimization / decision objective	Maximize internal validation macro-F1; recorded best value 0.8863293053632062
Comparator	Fixed standardized Euclidean uniform-vote KNN with k=5
Reported endpoints	Accuracy, macro/weighted F1, balanced accuracy, class metrics, confusion matrices and internal learning curves

Let $x_i \in \mathbb{R}^{36}$, $i = 1, \dots, n$, denote the current fitting records. For each attribute j, (1) estimates a separate mean and population standard deviation. These are attribute-specific quantities, rather than four statistics shared across patch positions.

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}\,, \qquad s_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \mu_j\right)^2}\,, \qquad j = 1, \dots, 36. \tag{1}$$

In (2), the fitted scale $\tilde{s}_j$ equals s_j when $s_j > 0$ and one when the attribute is constant. Validation and test records use the same fitting-estimated mean and scale. The standardized query z and standardized fitting record z_i are then compared using the Manhattan distance in (3).

$$z_j(x) = \frac{x_j - \mu_j}{\tilde{s}_j}, \qquad j = 1, \dots, 36. \tag{2}$$

$$d_1(z, z_i) = \sum_{j=1}^{36} \left| z_j - z_{ij} \right|. \tag{3}$$

For comparison, the baseline uses the Euclidean distance in (4), five neighbors and uniform weights. The proposed pipeline keeps $p = 1$ and inverse-distance voting fixed while selecting the neighbor count k.

$$d_2(z, z_i) = \sqrt{\sum_{j=1}^{36}\left(z_j - z_{ij}\right)^2}. \tag{4}$$

Let $\mathcal{N}_k(z)$ contain the indices of the k nearest fitting records under d_1, and write $d_i = d_1(z, z_i)$. Equation (5) reproduces the library's distance-weighting rule, including exact matches. The indicator $\mathbf{1}[P]$ is one when condition P holds and zero otherwise. If any selected neighbor has zero distance, only the zero-distance neighbors vote; otherwise each selected neighbor has weight $1/d_i$. No extra epsilon or adjustable exponent is introduced.

$$w_i(z) = \begin{cases} \mathbf{1}[d_i = 0], & \text{if } \min_{h \in \mathcal{N}_k(z)} d_h = 0, \\ 1/d_i, & \text{otherwise,} \end{cases} \qquad i \in \mathcal{N}_k(z). \tag{5}$$

For $\mathcal{L} = \{1,2,3,4,5,7\}$, the class score and final decision are defined in (6), where y_i is the fitting label. Equal class scores are resolved by the smallest class label, consistent with the classifier's ordered class representation. Normalized vote shares are not evaluated for probability calibration.

$$\begin{aligned} S_c(z) &= \sum_{i \in \mathcal{N}_k(z)} w_i(z)\,\mathbf{1}[y_i = c], \\ \hat{y}(z) &= \arg\max_{c \in \mathcal{L}} S_c(z). \end{aligned} \tag{6}$$

The internal validation criterion is the six-class macro-F1 in (7). Here TP_c, FP_c and FN_c are the validation true-positive, false-positive and false-negative counts for class c.

$$\mathrm{MacroF1} = \frac{1}{6}\sum_{c \in \mathcal{L}} \frac{2\mathrm{TP}_c}{2\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}. \tag{7}$$

Let $\mathcal{H}_8 \subset \{1, \dots, 35\}$ be the eight evaluated neighbor counts. The selected count is given in (8), with $\hat{y}_{\mathrm{val}}^{(k)}$ denoting the predictions of a candidate pipeline fitted only on the internal fitting records. The chosen count is frozen before the scaler and neighbor model are refitted on all official training observations. This criterion uses the evaluated candidates and does not imply adaptive TPE proposals during the eight-trial startup run.

$$k^* = \arg\max_{k \in \mathcal{H}_8} \mathrm{MacroF1}\left(y_{\mathrm{val}}, \hat{y}_{\mathrm{val}}^{(k)}\right). \tag{8}$$

Implementation and reproducibility. The notebook uses seed 42, FAST_MODE=True, a stratified 75:25 internal split, eight candidate evaluations, k in 1–35 and n_jobs=1. Both final pipelines are fitted on the full official training set. Learning curves use fitting subsets of 831, 1,663, 2,494 and 3,326 records and the same 1,109 validation records. The selected k remains fixed for the curve. Package versions, hardware details, raw-data hashes and a full trial history are not recorded in the supplied notebook. Repeated seeds, geographic-block validation, confidence intervals, calibration, latency and additional component ablations were not executed.

3. Results and Discussion

3.1. Overall Classification Performance

The two final KNN pipelines were fitted on the same 4,435 official training records and evaluated on the same 2,000 official test observations. Table 5 reports the saved comparison using macro-F1 as the principal classification endpoint, together with accuracy, weighted F1 and balanced accuracy.

Table 5. Official-test performance of the two standardized KNN configurations

Model	Accuracy (%)	Macro F1 (%)	Weighted F1 (%)	Balanced acc. (%)
Euclidean_KNN_Baseline	90.45	89.16	90.37	88.86
Weighted_Manhattan_KNN	90.85	89.64	90.80	89.38

Table 5 shows that Weighted_Manhattan_KNN achieved 90.85% accuracy and 89.64% macro-F1, compared with 90.45% and 89.16% for Euclidean_KNN_Baseline. The selected configuration correctly classified 1,817 observations, versus 1,809 for the baseline, an increase of eight correct predictions or 0.40 percentage points. The macro-F1 gain is 0.488944 percentage points, rounded to 0.49 points, when calculated from the unrounded metrics. The separately rounded percentages displayed in the table differ by 0.48 points. Weighted F1 increased from 90.37% to 90.80% and balanced accuracy from 88.86% to 89.38%.

These are descriptive differences for the fixed test partition. The experiment changes distance, vote weighting and k together, so it does not establish which individual change produced the gain. It also does not compare separately tuned Euclidean and Manhattan pipelines under equal search budgets.

3.2. Per-Class Performance and Confusion-Matrix Analysis

Table 6 presents precision, recall, F1 and support for each observed class, along with macro and weighted averages. The complete class table is necessary because the aggregate improvement is not uniform across categories.

Table 6. Complete per-class report for baseline and weighted Manhattan KNN

Class	Support	Base P	Base R	Base F1	Weighted P	Weighted R	Weighted F1
1	461	0.9786	0.9935	0.9860	0.9892	0.9892	0.9892
2	224	0.9775	0.9688	0.9731	0.9819	0.9688	0.9753
3	397	0.8894	0.9320	0.9102	0.8971	0.9219	0.9093
4	211	0.7286	0.6872	0.7073	0.7186	0.6777	0.6976
5	237	0.9200	0.8734	0.8961	0.9394	0.9156	0.9274
7	470	0.8766	0.8766	0.8766	0.8708	0.8894	0.8800
macro avg	2000	0.8951	0.8886	0.8916	0.8995	0.8938	0.8964
weighted avg	2000	0.9035	0.9045	0.9037	0.9078	0.9085	0.9080

Table 6 identifies class 4 as the weakest class under both configurations. Its recall decreases from 0.6872 to 0.6777, and its F1 decreases from 0.7073 to 0.6976. In contrast, class-5 recall increases from 0.8734 to 0.9156 and class-7 recall from 0.8766 to 0.8894. Class-2 recall remains unchanged, while recall for classes 1 and 3 decreases slightly. Thus, the higher macro average coexists with a lower recall for the most difficult class. Figure 3 displays the baseline confusion matrix as absolute counts and row-normalized proportions, using the class order 1, 2, 3, 4, 5 and 7.

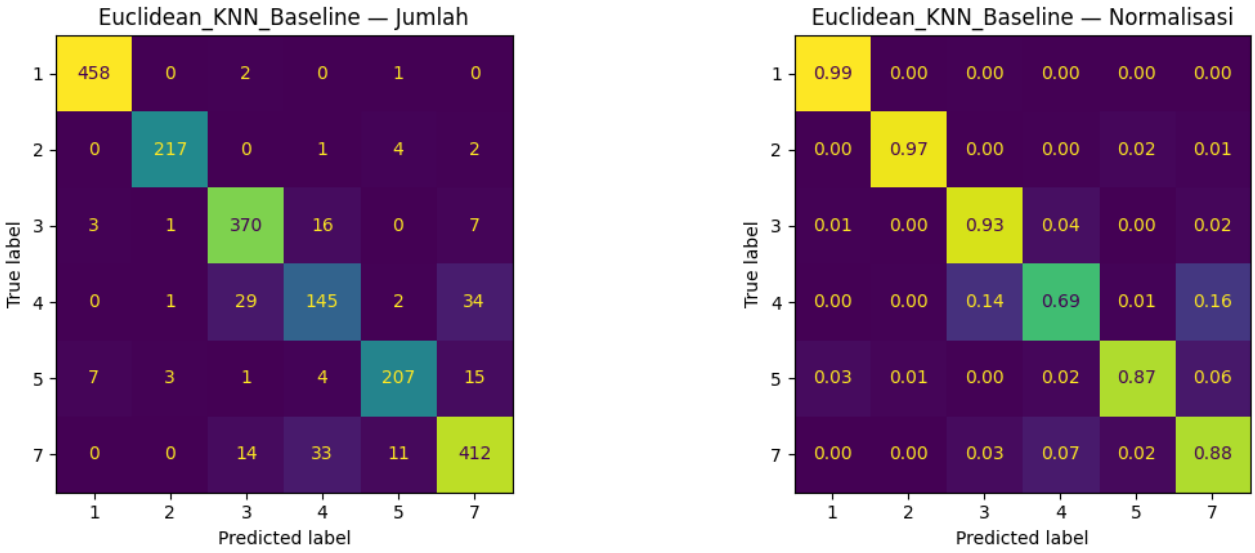


Figure 3. Euclidean KNN baseline confusion matrix: counts and row-normalized values

As Figure 3 shows, the baseline correctly predicts 145 of the 211 class-4 observations. Its main class-4 errors are assignments to class 7 (34 observations) and class 3 (29 observations). Class 5 has 207 correct predictions among

237 records. These counts provide the reference pattern for the selected configuration. Figure 4 presents the selected weighted Manhattan configuration on the same official test cases and with identical class ordering.

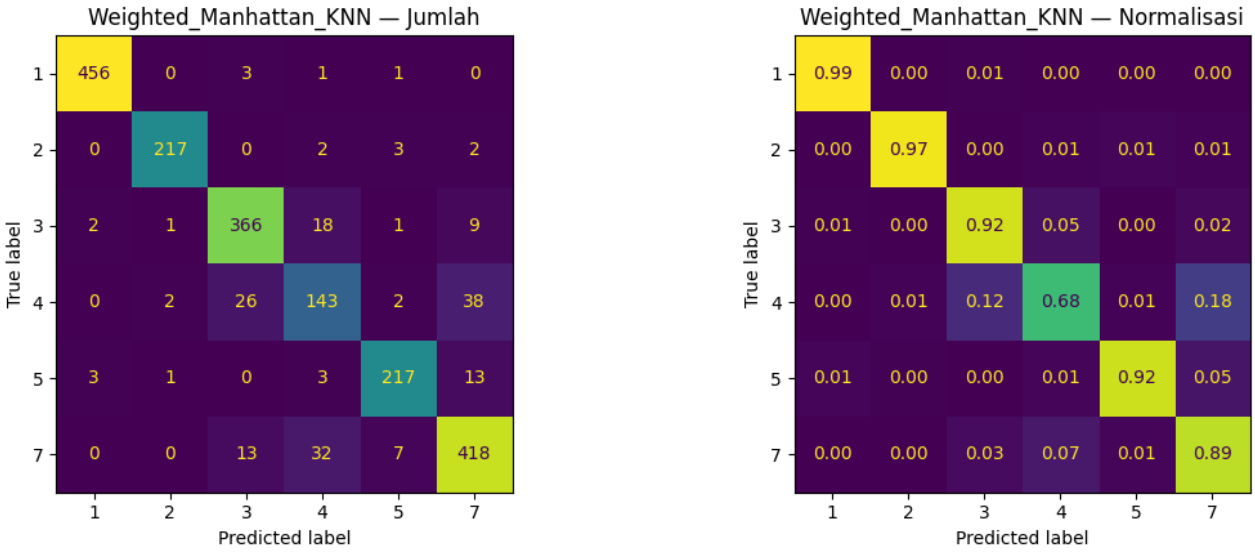


Figure 4. Weighted Manhattan KNN confusion matrix: counts and row-normalized values

Figure 4 shows 143 correct class-4 predictions, two fewer than the baseline, while class-4 errors assigned to class 7 increase from 34 to 38. Class-5 correct predictions rise from 207 to 217 and class-7 correct predictions from 412 to 418. The eight additional correct predictions overall therefore represent a redistribution of errors rather than universal improvement across classes.

3.3. Learning-Curve Assessment

Figure 5 compares training and validation macro-F1 at fitting sizes of 831, 1,663, 2,494 and 3,326 records. Both curves use the 1,109-record validation partition drawn from the official training set; the official test records are not involved. The selected k=3 remains fixed during these refits, and the scaler is refitted within each fitting subset.

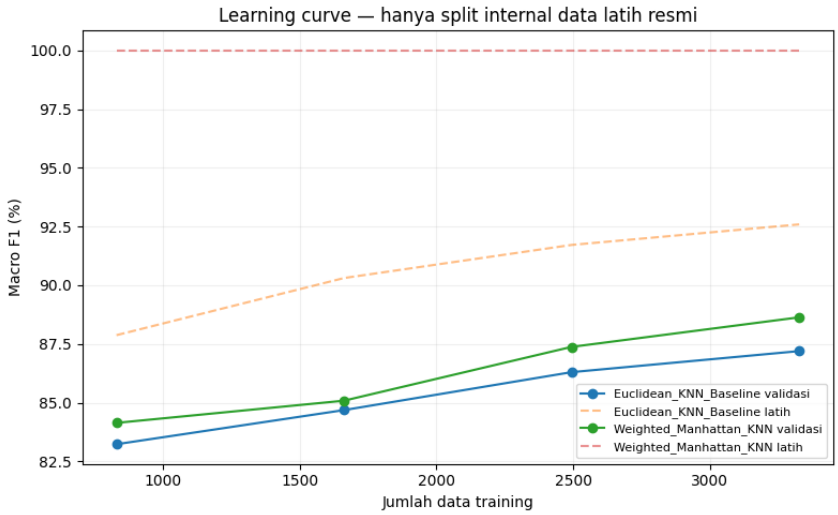


Figure 5. Training-only learning curves for the two KNN configurations

Figure 5 shows increasing validation scores as more training observations are used, with the weighted Manhattan curve above the baseline at the recorded sizes. The weighted model attains 100% training macro-F1 on these resubstitution predictions. This should be interpreted in the context of inverse-distance voting: a queried training record can match itself at zero distance, giving zero-distance matches effective control of the vote. The perfect training curve is therefore not a leave-one-out estimate or independent evidence of generalization. The validation partition also served to select k, so the curve is an internal diagnostic rather than an additional independent evaluation.

3.4. Discussion and Limitations

The results support a small observed aggregate gain for the selected standardized weighted Manhattan configuration under the fixed official partition. They do not isolate standardization, because both arms already use the same scaler procedure, and they do not isolate distance or weighting, because these change together with k. Separate unscaled, Manhattan-uniform and fixed-k weighting ablations were not executed. An independently adjustable distance exponent q was neither implemented nor searched.

The search contains eight candidate evaluations through TPESampler without an explicit startup setting. Under the documented ten-trial default, this budget does not reach model-guided TPE sampling, and the experiment does not establish an advantage of TPE over random search. Further limitations include one seed, one reused internal validation partition, unrecorded package and hardware details, and the absence of confidence intervals, calibrated probabilities, runtime measurements or independent geographic scenes. The increase on one official test set should not be described as statistical significance, globally optimal k, operational deployment readiness or geographic transferability.

4. Conclusion

This study compared two standardized KNN configurations on the official UCI Statlog Landsat Satellite split, retaining the same 36 original attributes and training-fitted StandardScaler. The Euclidean, uniform-vote baseline used k=5; the weighted Manhattan model selected k=3 from eight training-only candidate evaluations. Official-test accuracy increased from 90.45% to 90.85%, and macro-F1 from 89.16% to 89.64%. The macro-F1 gain calculated before rounding is 0.49 percentage points. Class-5 and class-7 recall improved, while class-4 recall declined from 68.72% to 67.77%. These results describe a configuration comparison with class-wise trade-offs. Distance, voting and k changed together, so their separate contributions remain unmeasured. The startup-only search and single split do not establish optimizer superiority, statistical stability or geographic transfer. Further work should compare components under equal budgets, extend the search beyond startup, record the environment, repeat seeds and evaluate independent scenes.

References

[1] I. Kotaridis and M. Lazaridou, “CNNs in land cover mapping with remote sensing imagery: a review and meta-analysis,” *Int. J. Remote Sens.*, vol. 44, no. 19, pp. 5896–5935, 2023, doi: 10.1080/01431161.2023.2255354.

[2] D. Moraes, M. L. Campagnolo, and M. Caetano, “Training data in satellite image classification for land cover mapping: a review,” *Eur. J. Remote Sens.*, vol. 57, no. 1, p. 2341414, 2024, doi: 10.1080/22797254.2024.2341414.

[3] Y. Wang, Y. Sun, X. Cao, Y. Wang, W. Zhang, and X. Cheng, “A review of regional and Global scale Land Use/Land Cover (LULC) mapping products generated from satellite remote sensing,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 206, pp. 311–334, 2023, doi: 10.1016/j.isprsjprs.2023.11.014.

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

[5] P. Dou, H. Shen, C. Huang, Z. Li, Y. Mao, and X. Li, “Large-scale land use/land cover extraction from Landsat imagery using feature relationships matrix based deep-shallow learning,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 129, p. 103866, 2024, doi: 10.1016/j.jag.2024.103866.

[6] Y. Maulani and K. Surendro, “Detailed land use classification model based on vegetation indices and texture features,” *Remote Sens. Appl.*, vol. 40, p. 101786, 2025, doi: 10.1016/j.rsase.2025.101786.

[7] V. Sangeetha and L. Agilandeeswari, “Artificial intelligence enabled spectral-spatial feature extraction techniques for land use and land cover classification using hyperspectral images – An inclusive review,” *The Egyptian Journal of Remote Sensing and Space Sciences*, vol. 28, no. 3, pp. 455–467, 2025, doi: 10.1016/j.ejrs.2025.06.004.

[8] A. H. Alshammari, G. Bencsik, and A. H. Ali, “A Survey of Six Classical Classifiers, Including Algorithms, Methodological Characteristics, Foundational Variants, and Recent Advances,” *Algorithms*, vol. 19, no. 1, p. 37, 2026, doi: 10.3390/a19010037.

[9] R. K. Halder, M. N. Uddin, Md. A. Uddin, S. Aryal, and A. Khraisat, “Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications,” *J. Big Data*, vol. 11, no. 1, p. 113, 2024, doi: 10.1186/s40537-024-00973-y.

[10] L. B. V de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” *Appl. Soft Comput.*, vol. 133, p. 109924, 2023, doi: 10.1016/j.asoc.2022.109924.

[11] O. U. Lenz, H. Bollaert, and C. Cornelis, "A unified weighting framework for evaluating nearest neighbour classification," *Fuzzy Sets Syst.*, vol. 519, p. 109516, 2025, doi: 10.1016/j.fss.2025.109516.
[12] F. Alonso-Sarría, C. Valdivieso-Ros, and F. Gomariz-Castillo, "Analysis of the hyperparameter optimisation of four machine learning satellite imagery classification methods," *Comput. Geosci.*, vol. 28, no. 3, pp. 551–571, 2024, doi: 10.1007/s10596-024-10285-y.
[13] M. Baratchi *et al.*, "Automated machine learning: past, present and future," *Artif. Intell. Rev.*, vol. 57, no. 5, p. 122, 2024, doi: 10.1007/s10462-024-10726-1.
[14] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1484, 2023, doi: 10.1002/widm.1484.
[15] A. Apicella, F. Isgrò, and R. Prevete, "Don't push the button! Exploring data leakage risks in machine learning and transfer learning," *Artif. Intell. Rev.*, vol. 58, no. 11, p. 339, 2025, doi: 10.1007/s10462-025-11326-3.
[16] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023, doi: 10.1016/j.patter.2023.100804.
[17] M. A. Lones, "Avoiding common machine learning pitfalls," *Patterns*, vol. 5, no. 10, p. 101046, 2024, doi: 10.1016/j.patter.2024.101046.
[18] S. Friedrich and T. Friede, "On the role of benchmarking data sets and simulations in method comparison studies," *Biometrical Journal*, vol. 66, no. 1, p. 2200212, 2024, doi: 10.1002/bimj.202200212.
[19] S. Kapoor *et al.*, "REFORMS: Consensus-based Recommendations for Machine-learning-based Science," *Sci. Adv.*, vol. 10, no. 18, p. eadk3452, 2024, doi: 10.1126/sciadv.adk3452.
[20] J. L. J. Pereira, K. Smith-Miles, M. A. Muñoz, and A. C. Lorena, "Optimal selection of benchmarking datasets for unbiased machine learning algorithm evaluation," *Data Min. Knowl. Discov.*, vol. 38, no. 2, pp. 461–500, 2024, doi: 10.1007/s10618-023-00957-1.
[21] D. Abriha, P. K. Srivastava, and S. Szabó, "Smaller is better? Unduly nice accuracy assessments in roof detection using remote sensing data with machine learning and k-fold cross-validation," *Heliyon*, vol. 9, no. 3, p. e14045, 2023, doi: 10.1016/j.heliyon.2023.e14045.
[22] S. de Bruin, D. J. Brus, G. B. M. Heuvelink, T. van Ebbenhorst Tengbergen, and A. M. J.-C. Wadoux, "Dealing with clustered samples for assessing map accuracy by cross-validation," *Ecol. Inform.*, vol. 69, p. 101665, 2022, doi: 10.1016/j.ecoinf.2022.101665.
[23] A. Stock, "Choosing blocks for spatial cross-validation: lessons from a marine remote sensing case study," *Frontiers in Remote Sensing*, vol. 6, p. 1531097, 2025, doi: 10.3389/frsen.2025.1531097.
[24] Y. Wang, M. Khodadadzadeh, and R. Zurita-Milla, "Spatial+: A new cross-validation method to evaluate geospatial machine learning models," *International Journal of Applied Earth Observation and Geoinformation*, vol. 121, p. 103364, 2023, doi: 10.1016/j.jag.2023.103364.
[25] S. Farhadpour, T. A. Warner, and A. E. Maxwell, "Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices," *Remote Sens. (Basel).*, vol. 16, no. 3, p. 533, 2024, doi: 10.3390/rs16030533.
[26] L. Maier-Hein *et al.*, "Metrics reloaded: recommendations for image analysis validation," *Nat. Methods*, vol. 21, no. 2, pp. 195–212, 2024, doi: 10.1038/s41592-023-02151-z.
[27] J. Opitz, "A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 820–836, 2024, doi: 10.1162/tacl_a_00675.
[28] A. Reinke *et al.*, "Understanding metric-related pitfalls in image analysis validation," *Nat. Methods*, vol. 21, no. 2, pp. 182–194, 2024, doi: 10.1038/s41592-023-02150-0.
[29] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in Deploying Machine Learning: A Survey of Case Studies," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–29, 2023, doi: 10.1145/3533378.
[30] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: a survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112, no. 9, pp. 3211–3260, 2023, doi: 10.1007/s10994-023-06336-7.
[31] S. Bates, T. Hastie, and R. Tibshirani, "Cross-Validation: What Does It Estimate and How Well Does It Do It?," *J. Am. Stat. Assoc.*, vol. 119, no. 546, pp. 1434–1445, 2024, doi: 10.1080/01621459.2023.2197686.
[32] T. Viering and M. Loog, "The Shape of Learning Curves: A Review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7799–7819, 2023, doi: 10.1109/tpami.2022.3220744.
[33] H. Andalib, H. Ebadi, and R. Naanjam, "An intelligent dual-stage fusion framework of optical and radar data for land cover classification," *Remote Sens. Appl.*, vol. 42, p. 101987, 2026, doi: 10.1016/j.rsase.2026.101987.
[34] M. R. Islam, A. A. Lima, S. C. Das, M. F. Mridha, A. R. Prodeep, and Y. Watanobe, "A Comprehensive Survey on the Process, Methods, Evaluation, and Challenges of Feature Selection," *IEEE Access*, vol. 10, pp. 99595–99632, 2022, doi: 10.1109/access.2022.3205618.

[35] P. Koukaras and C. Tjortjis, "Data Preprocessing and Feature Engineering for Data Mining: Techniques, Tools, and Best Practices," *AI*, vol. 6, no. 10, p. 257, 2025, doi: 10.3390/ai6100257.

[36] H. İ. Gündüz, O. Orhan, S. N. G. Hamal, and S. Ekercin, "An Explainable and Optimized Machine Learning Framework for Multisensor Remote Sensing-Based Land Use and Land Cover Classification," *Transactions in GIS*, vol. 30, no. 3, p. e70293, 2026, doi: 10.1111/tgis.70293.

[37] A. Candelieri, A. Ponti, and F. Archetti, "Fair and green hyperparameter optimization via multi-objective and multiple information source Bayesian optimization," *Mach. Learn.*, vol. 113, no. 5, pp. 2701–2731, 2024, doi: 10.1007/s10994-024-06515-0.

[38] A. Morales-Hernández, I. Van Nieuwenhuyse, and S. Rojas Gonzalez, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8043–8093, 2023, doi: 10.1007/s10462-022-10359-2.

[39] D. Valle, R. Leite, R. Izbicki, C. Silva, and L. Haneda, "Local uncertainty maps for land-use/land-cover classification without remote sensing and modeling work using a class-conditional conformal approach," *International Journal of Applied Earth Observation and Geoinformation*, vol. 135, p. 104288, 2024, doi: 10.1016/j.jag.2024.104288.

[40] A. Comber and N. Tsutsumida, "Geographically weighted accuracy for hard and soft land cover classifications: 5 approaches with coded illustrations," *Int. J. Remote Sens.*, vol. 44, no. 19, pp. 6233–6257, 2023, doi: 10.1080/01431161.2023.2264503.

[41] C. Wang, R. Shang, Y. Li, J. Feng, and S. Xu, "Cross-scene hyperspectral image classification based on cross-domain feature extraction and category decision collaborative optimization," *Expert Syst. Appl.*, vol. 296, p. 128842, 2026, doi: 10.1016/j.eswa.2025.128842.

[42] X. Zhang and B. Liu, "Open-Set Cross-Domain Hyperspectral Image Classification Based on Manifold Mapping Alignment," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 6241–6252, 2024, doi: 10.1109/jstars.2024.3396883.