Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Near-Duplicate Contamination Is Concentrated, Not Uniform: A Class- Level Audit of a Public Banana Nutrient-Deficiency Image Dataset and Its Effect on Benchmark Accuracy

Yogesh Yogesh

yogesh-aiml@dsu.edu.in

Dayananda Sagar University

P Sudir

Dayananda Sagar University

Princy Randhawa

Dayananda Sagar University

Muzameel Ahmed

Dayananda Sagar Academy of Technology and Management

Chandrakala B M

Dayananda Sagar College of Engineering

Article

Keywords: data leakage, near-duplicate detection, nutrient deficiency, precision agriculture, benchmark evaluation, reproducibility, convolutional neural networks

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11090762/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Near-Duplicate Contamination Is Concentrated, Not Uniform: A Class-Level Audit of a Public Banana Nutrient-Deficiency Image Dataset and Its Effect on Benchmark Accuracy

Yogesh[1*], Sudir Prakash[2], Princy Randhawa[3]

[1,2,3] Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), School of Engineering, Dayananda Sagar University, Harohalli Campus, Bengaluru 562112, Karnataka, India

yogesh-aiml@dsu.edu.in[1], sudir-aiml@dsu.edu.in[2], princy-aiml@dsu.edu.in[3]

ORCID: 0000-0002-0746-0558[1], 0009-0000-8334-7365[2], 0000-0001-6356-8718[3]

*Corresponding author: yogesh-aiml@dsu.edu.in

Abstract

A random split of a single public dataset is used to score the majority of nutrient-deficiency classifiers. That is only effective if every photograph is a unique specimen. Frequently, it isn't. The score partially assesses recollection of previously seen content because augmented copies are often published alongside their originals, and duplicates of one leaf are randomly dropped into training and test alike. Using two perceptual hashes and a grouping strategy based on seeds that prevents unrelated photos from being chained into a single cluster, we audited all 10,600 images of a banana deficiency collection. Outcome: 4,155 photos, 2,005 groups, and 39.2% of the release. No group held more than six images, and 1,882 of them bridged the raw and augmented folders. That headline number turns out to describe the dataset badly. Every image in the healthy class and every image in the sulphur class sits inside a duplicate group; the other seven classes average 11.0%. Contamination under the 70:10:20 split used for training reaches 27.1% of test images overall—71.6% in healthy, 66.4% in sulphur, and below 11% everywhere else. Six matched runs of MobileNetV3-Large gave 88.45% (SD 0.57) on the random split against 82.73% (SD 0.85) on the group-aware one, a shortfall of 5.72 points (Welch $t = 13.7$, $p < 0.001$, 95% CI 4.8 to 6.7). Per-class recall locates it. Sulphur falls 14.3 points and healthy 10.0; no other class loses more than 3.5, while three classes improve, potassium by 9.0. Scoring one trained model separately on the contaminated and clean halves of its own test set gave 92.87% against 85.05%, a 7.82-point gap ($p = 1.7 \times 10^{-6}$). Balanced accuracy went the other way, 83.42% against 85.40%, because 79% of the contaminated half comes from the two affected classes. Contamination here is real, and it costs accuracy, but it belongs to particular classes rather than to the dataset, and a single percentage is a poor guide to it. The audit tool, group listings, and corrected partitions are released.

Keywords: data leakage; near-duplicate detection; nutrient deficiency; precision agriculture; benchmark evaluation; reproducibility; convolutional neural networks

1. Introduction

Leaf photography is now a standard route to crop diagnosis. The accuracies reported for it are high—above ninety-five percent has been unremarkable since the early demonstrations that convolutional networks could separate dozens of plant diseases from photographs [1], [2]. Read plainly, such numbers say the problem is finished.

Anyone who has carried one of these models into an orchard knows better. The gap between benchmark and deployment has been documented for a decade [3], [4]. Distribution shift is the standard explanation, and it is a good one: benchmark photographs isolate a single leaf under even light, while field photographs contain soil, sky, neighboring plants, and whatever the weather is doing. Note where that explanation puts the difficulty. Entirely outside the benchmark.

A second difficulty sits inside it. Agricultural datasets are often published after augmentation or hold repeat exposures of one specimen, or both. Split such a collection at random, and copies of a single leaf land on both sides of the partition. A model can then score well by recognizing a leaf it has already met rather than by reading

the symptom. Kaufman and colleagues formalized this as leakage [5]; Kapoor and Narayanan later traced a good deal of irreproducible science back to it [6].

General-purpose benchmarks have been checked for this. Barz and Denzler put near-duplicates at roughly 3.3% of CIFAR-10 and 10% of CIFAR-100 test images, and removing them shuffled the model rankings [7]. Fresh test sets built for CIFAR-10 and ImageNet cost several accuracy points [8]. Label errors turned up across the test sets of ten standard benchmarks [9]. No equivalent measurement exists for an agricultural leaf collection, so far as we can establish — which is odd, since publishing in augmented form makes the problem more likely, not less.

1.1 Research questions

1. How much of a representative public nutrient-deficiency dataset consists of near-duplicate images, and is that duplication spread evenly across classes?
2. Under the partitioning protocol actually used for training, what share of the test set has a counterpart in training?
3. How far does that contamination lift the reported accuracy of a standard classifier, and can the lift be attributed to contamination rather than to some other consequence of changing the split?

1.2 Contributions and deliverables

What follows is a measurement, not a method. The deliverables:

- A class-level audit showing that duplication in this collection is confined almost entirely to two of its nine classes, a structure invisible in the aggregate figure that a whole-dataset audit would report.
- A grouping procedure (Algorithm 1) immune to the transitive-chaining collapse that afflicts union-find clustering on uniformly resized imagery, together with an automatic diagnostic for that failure.
- Contamination measured under the same 70:10:20 protocol used for training, rather than under a convenient approximation of it.
- A control experiment (Section 5.4) in which a single trained model is scored on the contaminated and clean parts of its own test set, holding training data fixed and thereby separating contamination from the change in training composition that group-aware splitting also brings.
- Per-class evidence locating the accuracy loss in the two classes the audit flags, rather than asserting a dataset-wide effect.
- Public release of the audit tool, duplicate group listings, and group-aware partitions.

We are not arguing that the dataset is unfit for use, nor that published work on it is wrong. The point is narrower. Accuracies taken from random splits of this collection carry an upward bias of roughly six points—larger than the margins that usually separate competing methods — and that bias is concentrated, not spread.

1.3 Organisation

Section 2 covers deep learning for plant disease and deficiency, the benchmark-to-field literature, leakage and evaluation integrity, and near-duplicate detection, then places this study among them. Section 3 contains the computing environment and dataset. Section 4 describes the methodology in the order it was carried out: near-duplicate detection (Algorithm 1), visual verification, group-aware partitioning (Algorithm 2), contamination measurement, the training protocol, and the statistics. Results are presented in Section 5, research question by research question, with the per-class breakdown at 5.5 and the control experiment at 5.4. Section 6 discusses the implications for evaluation practice. Section 7 contains limitations; Section 8 contains conclusions and future work.

2. Related Work

2.1 Deep learning for nutrient deficiencies and plant diseases

The field's benchmark is PlantVillage [10], along with the findings of Mohanty and associates that refined AlexNet and GoogLeNet to differentiate 26 illnesses across 14 species at better than 99% on a held-out split [1]. Ferentinos performed the same exercise on additional designs and obtained essentially the same result [2]. The fine-tuning method was then the subject of comparative research [11], [12].

Classification of deficiencies grew relatively slowly. The research has been shaped by the lack of labeled data: rice nitrogen status [13], apple deficit [14], fruit defects linked to deficiencies [15], and calcium insufficiency in pears [16]. The banana collection studied here was published by Sunitha and colleagues [17], and it has been used in subsequent categorization investigations.

2.2 Benchmark and field

Doubt arrived early. Barbedo catalogued the obstacles to automatic disease identification [3], then demonstrated that variety in the training set matters more than its size when transfer is at stake [4], and later that working from individual lesions instead of whole leaves changes the problem substantially [18]. PlantDoc exists for exactly this reason—field imagery set against PlantVillage's laboratory conditions—and models trained on the latter transferred poorly to the former [19].

Distribution shift is a genuine obstacle, then. But none of that work asks whether the benchmark figure itself was measured correctly, which is our question. The two compound. Quote a model at 99% on a contaminated benchmark, and its true benchmark score is lower, and its field score is lower again.

2.3 Leakage and evaluation integrity

Leakage has a standard definition from Kaufman and colleagues—information about the target that would not be available when the prediction is made—along with a catalogue of its varieties [5]. Seventeen fields surveyed by Kapoor and Narayanan showed the same pattern behind irreproducible results [6]. Medical imaging offers the sharpest case: of several hundred COVID-19 models reviewed by Roberts and colleagues, none was judged fit for clinical use, and duplicated or badly partitioned data appeared repeatedly among the faults [20]. Varoquaux and Cheplygina found much the same across medical imaging [21], as did Whalen and colleagues in genomics [22].

Barz and Denzler is our nearest methodological precedent [7]—near-duplicates located by feature-space similarity, a cleaned test set published, and rankings shown to move. Two differences here. We use perceptual hashing rather than learned features, for reasons given in Section 2.4, and we work in a setting where shipping augmented copies pushes contamination an order of magnitude above what CIFAR shows.

2.4 Near-duplicate detection

A perceptual hash reduces an image to a short binary code, arranged so that similar images produce codes a small Hamming distance apart. Both the difference hash and the average hash are long-established [23]. Embeddings from a pretrained network are the obvious alternative; they capture semantic rather than surface similarity. Surface similarity is what we want. The target is the same photograph lightly altered, not a picture of something comparable. Hashing has two further merits: it runs identically every time, and it needs no model, which matters when the whole point is that somebody else can repeat the exercise.

2.5 Position of this study

Table 1 places this work among the closest comparisons. We know of no earlier study that quantifies near-duplicate contamination in an agricultural leaf collection, checks the detection by eye, breaks it down by class, and measures the consequence under controlled conditions.

Table 1. The study against the nearest comparable work.					
Study	**Domain**	**Contamination quantified**	**Visual check**	**Class-level**	**Accuracy effect**
Mohanty et al. [1]	Plant disease	No	No	No	No
Barbedo [4]	Plant disease	No	No	No	Partial
Recht et al. [8]	General vision	Indirect	No	No	Yes
Barz & Denzler [7]	General vision	Yes (3–10%)	Partial	No	Yes
Northcutt et al. [9]	General vision	Label errors	Yes	Partial	Yes
Kapoor & Narayanan [6]	Cross-field	Survey	No	No	No
This study	Nutrient deficiency	Yes (39.2%)	Yes	Yes	Yes (5.72 pp)

3. Materials

3.1 Dataset

The collection is the banana nutrient-deficiency dataset of Sunitha and colleagues, described in Data in Brief and distributed through Mendeley Data [17]. It covers nine classes—boron, calcium, iron, magnesium, manganese, potassium, sulphur, and zinc deficiency, plus healthy. Photographs were taken at plantations across Karnataka, India, under whatever daylight was available. Six cultivars are covered: Dwarf Cavendish, Robusta, Rasthali, Poovan, Monthan, and Elakkibale. Cultivar appears in neither the folder names nor the filenames, which rules out any cultivar-wise analysis.

Backgrounds were removed by the dataset authors, who also cropped by hand and resized every image to 225 × 225 pixels, so every distributed image shows one leaf on plain black. The data article reports roughly 3,000 raw images, expanded to more than 7,000 by augmentation [17]. We analysed version 1 of the Mendeley record, downloaded on 13 September 2026. That download contains 10,600 JPEG images in two top-level folders, 3,100 in the raw folder and 7,500 in the augmented folder. These counts are consistent with the approximate figures in the data article, which reports the augmented total as “7000+” rather than an exact number, and every result below refers to the files actually present in that download. Figure 1 and Table 2 give the class breakdown.

The data article also notes that a leaf may be deficient in more than one nutrient, so the same leaf image can appear in more than one class folder [17]. Multi-folder placement is therefore a documented feature of the release rather than an annotation error. It is a second route, alongside augmentation, by which one specimen can reach both sides of a random partition, and Section 7 returns to it.

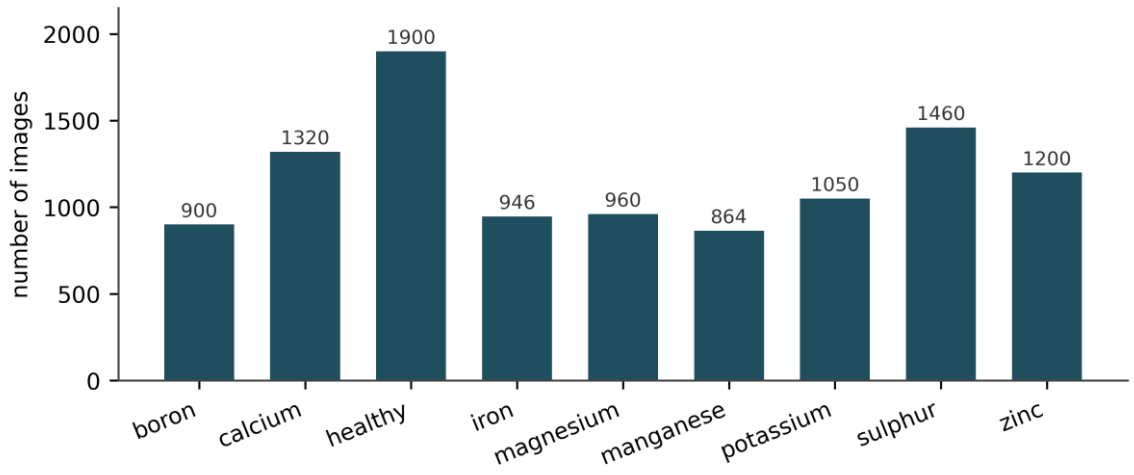

Figure 1. Class distribution as published. Healthy is the largest class at 1,900 images, manganese the smallest at 864, an imbalance of roughly 2.2 to 1.

Table 2. Class composition and duplication.

Class	Images	Share (%)	In duplicate group	Duplicated (%)	Test contaminated (%)
boron	900	8.5	94	10.4	8.0
calcium	1,320	12.5	149	11.3	9.8
healthy	1,900	17.9	1,900	100.0	71.6
iron	946	8.9	132	14.0	8.7
magnesium	960	9.1	89	9.3	9.0
manganese	864	8.2	122	14.1	10.7
potassium	1,050	9.9	114	10.9	7.3
sulphur	1,460	13.8	1,460	100.0	66.4
zinc	1,200	11.3	95	7.9	5.0
Total	10,600	100.0	4,155	39.2	27.1

The final column gives the share of that class's test images having a near-duplicate in training under the 70:10:20 partition.

3.2 Computing environment

Audit and partitioning ran on two CPU cores, about fifteen minutes for the whole collection. Training used a single NVIDIA T4 in a hosted notebook, roughly forty-five minutes a run. Python 3.12 throughout, with PyTorch and torchvision for training and Pillow with NumPy for the audit. No proprietary data, no laboratory, no fieldwork, no funding.

4. Methodology

Figure 2 lays out the procedure: audit the collection for near-duplicate groups, check those groups by eye, build two partitions from the same images, and train one classifier under each.

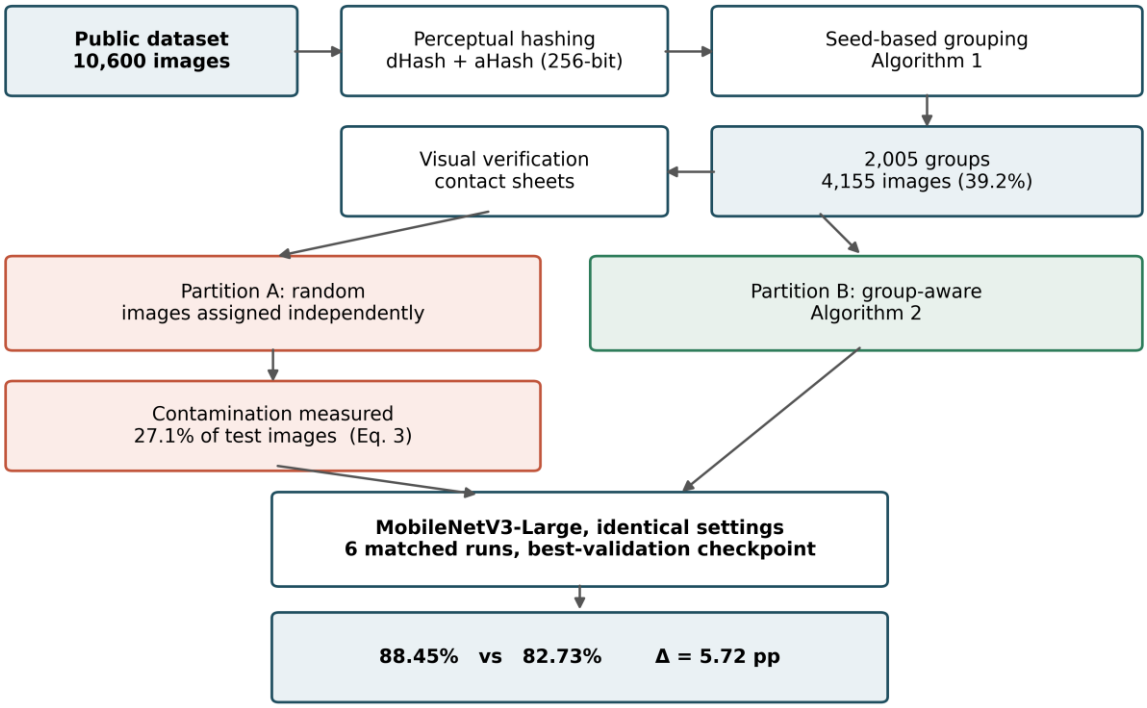

Figure 2. Experimental pipeline. Partition is the only thing that differs between the two training conditions; model, hyperparameters, augmentation, and test-set size are held fixed.

4.1 Finding near-duplicates

Three signals were combined. MD5 caught byte-identical files. Filenames were checked for transformation markers—rotation or flip suffixes, which in some releases let an augmented copy be traced straight back to its source—but this collection carries none, so that signal gave us nothing. That work was left entirely to perceptual hashing.

Let I denote an image, and let G(I) denote its conversion into grayscale to make a grid of (s + 1) × s, where s = 16. The difference hash is interpreted as having just one bit for any adjacent horizontal pair [23]:

$$h^d(I)_k = 1 \text{ if } G(I)_{i,j+1} > G(I)_{i,j}\,,\ 0 \text{ otherwise},\quad k = i{\cdot}s + j,\quad i, j = 1 \ldots s \tag{1}$$

which comes to 256 bits. The average hash $h^a(I)$ is built the same way over an s × s reduction, each bit recording whether its pixel sits above the mean. Both codes must agree inside their thresholds for a pair to count as near-duplicate:

$$D(I_1, I_2) = 1 \iff d_h(h^d(I_1), h^d(I_2)) \le \tau^d \ \wedge\ d_h(h^a(I_1), h^a(I_2)) \le \tau^a \tag{2}$$

Here d_h is Hamming distance, $\tau^d = 12$ and $\tau^a = 16$ out of 256—agreement on 95.3% and 93.8% of bits.

Because h^d is capable of reading local gradient structure while h^a can read global intensity, the requirement of constructing two different hashes reduces the chances of spurious matching. The code length is not arbitrarily selected. A first attempt with 64-bit hashes produced heavy collisions on this data, because every image shares the same dimensions and a black background.

4.1.1 Seed-based grouping

The groups are built around seeds, not transitively. The natural union-find implementation merges A with C whenever both happen to match B, even where A and C share nothing. On uniformly resized imagery, that chain can run away, and we watched it do so. Applied during development to a field-photograph collection of banana images, transitive grouping returned one group of 4,593 images—17.2% of that dataset—which, on inspection, held visibly different plants.

Algorithm 1 gives every group a single seed and admits a candidate only on a direct match to it. Chaining then cannot happen. A warning also fires whenever the largest group passes two percent of the collection. It stayed silent here, and the biggest group held six images. That last fact is the strongest evidence we have that the detector is not over-merging: over-merging announces itself in enormous groups, not in hundreds of small tidy ones.

Algorithm 1: Seed-based near-duplicate grouping

Input : images $I = \{I_1 \dots I_n\}$; thresholds τ^d, τ^a
Output: disjoint groups $G = \{G_1 \dots G_m\}$

```
S ← ∅                          // seed list
for each I_i ∈ I do
    compute h^d(I_i), h^a(I_i)        // Eq. (1)
    C ← seeds sharing a 32-bit hash prefix bucket with I_i
    best ← ∅ ;  bestDist ← ∞
    for each s ∈ C do
        if D(I_i, I_s) = 1 and d_h(h^d(I_i), h^d(I_s)) < bestDist then
            best ← s ;  bestDist ← d_h(h^d(I_i), h^d(I_s))     // Eq. (2)
    if best ≠ ∅ then  add I_i to the group of best
    else              S ← S ∪ {I_i} ;  start a group seeded by I_i
merge any groups sharing an MD5 digest
if max|G_j| > 0.02·n then  raise over-merge warning
return G
```

4.2 Checking by eye

Software counts are not evidence on their own. The detector may be merging things a person would keep apart, so groups were rendered as contact sheets—up to 24 members tiled into one figure, every thumbnail labelled by class and source folder. We produced sheets for the largest groups, for a random sample of ordinary ones, and for any group whose members disagreed about class. Each sheet was put to a single question. Same leaf, or several leaves that happen to resemble one another?

4.3 Building the partitions

Two partitions were built from the same 10,600 images, each 70:10:20 across training, validation, and test.

- Partition A, random. Each image is assigned independently. This is what most published work on collections of this kind does, and it is the contaminated baseline.
- Partition B, group-aware. Duplicate groups are indivisible and go whole to one partition, following Algorithm 2, so no test image has a counterpart in training.

Test sets come out equal at 2,120 images apiece. Training sets differ by one image, 7,419 against 7,420, so training-set size cannot explain any gap. Composition is another matter, and we return to it in Section 5.4.

Algorithm 2: Group-aware partitioning

Input : groups $G = \{G_1 \dots G_m\}$; fractions f_test, f_val; seed σ
Output: $\pi(I) \in \{\text{train, val, test}\}$ for every image

```
N ← Σ_j |G_j|
shuffle G with seed σ
c ← 0
for each G_j ∈ G do
    r ← c / N
    if      r < f_test          then π(I) ← test   ∀ I ∈ G_j
    else if r < f_test + f_val  then π(I) ← val    ∀ I ∈ G_j
    else                             π(I) ← train  ∀ I ∈ G_j
    c ← c + |G_j|
return π
```

4.4 Measuring contamination

Contamination under Partition A was counted directly, no model involved. With T the test set, R the training set, and g(I) the group holding image I,

$$\rho = |\{ I \in T : \exists J \in R,\ g(J) = g(I) \}| \,/\, |T| \qquad (3)$$

which is the share of test images having at least one counterpart in training. Only the data and the partitioning rule enter into it. We report the figure under the same 70:10:20 split used for training rather than under a convenient approximation, because how large the training set is determines how many counterparts can fall inside it.

4.5 Model and training

MobileNetV3-Large [24] started from ImageNet weights [25], with the final layer replaced for nine outputs. It was picked as representative of the lightweight models proposed for agricultural deployment; whether the effect survives a change of architecture is left to Section 7. Images went in at 224 × 224, normalized with ImageNet statistics. Augmentation during training: horizontal and vertical flips, rotation to twenty degrees, and color jitter of 0.2 in brightness and contrast and 0.15 in saturation. AdamW [26] at 3×10^{-4}, cosine annealing, batch size 32, cross-entropy loss, 30 epochs. Validation accuracy was measured each epoch, and the best epoch's weights were restored before testing, so nothing hung on where training happened to stop. Settings were identical across conditions. Only the partition differs. Table 3 collects it all.

Table 3. Training configuration, identical between Partition A and Partition B

Parameter	Value	Reason
Backbone	MobileNetV3-Large [24]	Representative lightweight model
Initialisation	ImageNet pretrained [25]	Standard transfer learning
Input	224 × 224 px	Backbone native resolution
Optimiser	AdamW [26]	Decoupled weight decay
Learning rate	3×10^{-4}	Tuned on validation, Partition A
Schedule	Cosine annealing, T = 30	Removes epoch-count sensitivity
Batch size	32	GPU memory
Epochs	30	Validation plateau
Selection	Best validation epoch	Guards against late overfitting
Augmentation	flips, ±20°, colour jitter	Identical across conditions
Matched runs	6	Supports significance testing

4.6 Metrics and statistics

We report overall accuracy and balanced accuracy together. Balanced accuracy is the unweighted mean of per-class recall [27]:

$$BA = (1/K) \sum_k \left(TP_k / (TP_k + FN_k) \right), \quad K = 9 \qquad (4)$$

The two measures part company when errors fall unevenly across classes. That divergence turns out to carry the main message here. Conditions are compared with Welch's unequal-variance t-test.

$$t = (\bar{x}^A - \bar{x}^B) / \sqrt{(s^{A2}/n^A + s^{B2}/n^B)}, \quad n^A = n^B = 6 \tag{5}$$

An unpaired test is the right choice. Runs under the two conditions share no linking variable beyond their index, and treating them as matched pairs would overstate precision. Cohen's d gives the effect size, and a 95% confidence interval accompanies the difference. Section 5.4 scores one classifier on two disjoint subsets, which calls for something else again: a two-proportion z-test with a percentile bootstrap interval over 10,000 resamples.

Individual runs cannot be reproduced bit-for-bit, since GPU convolution algorithm selection is nondeterministic by default. Results are therefore means and standard deviations over independent runs.

5. Results

5.1 How much duplication, and where (RQ1)

2,005 groups of more than one image, covering 4,155 images, 39.2% of the collection. The largest held six. Algorithm 1's over-merge diagnostic did not fire.

Of those groups, 1,882 straddle the raw and augmented folders. That identifies the mechanism: augmented copies of raw originals, not repeat photography. Matching ran 63.1% of the 3,100 raw images to at least one augmented image and 25.8% of the 7,500 augmented images back to a raw original (Figure 3). Anything altered beyond what hashing can follow goes unrecovered, so treat these as lower bounds.

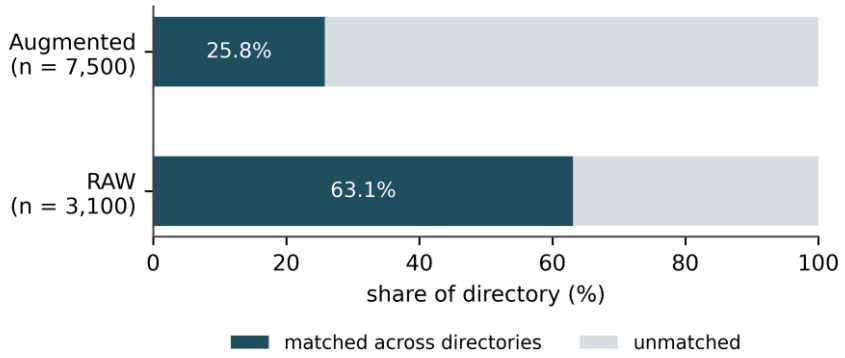

Figure 3. Share of each folder matched to the other. Unmatched augmented images were transformed past the reach of perceptual hashing, so the duplication reported here understates the true figure.

The aggregate hides the shape of the thing. Break it down by class (Figure 4, Table 2), and duplication turns out not to be distributed at all. Every image in the healthy class belongs to a duplicate group. So does every image in the sulphur class. The remaining seven sit between 7.9% and 14.1%, averaging 11.0%. Two classes thus account for 3,360 of the 4,155 duplicated images.

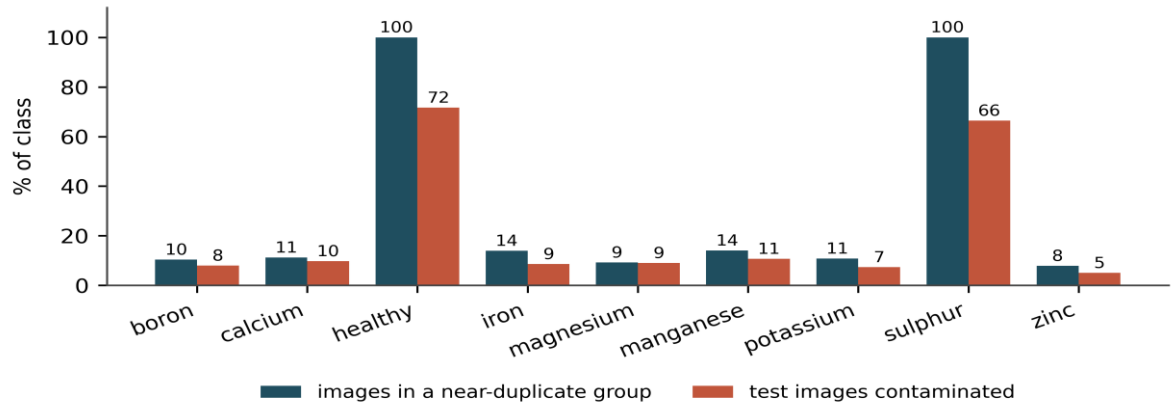

Figure 4. Duplication and contamination by class. Healthy and sulphur are wholly composed of duplicate groups; the other seven classes average 11.0% duplication and under 11% contamination.

Contact sheets of the largest groups showed one leaf repeated under changes of brightness, orientation, or scale, members drawn from both folders. RQ1 answers, then, as follows: duplication is extensive but concentrated, and a dataset-level percentage describes this collection poorly.

5.2 Contamination under the training protocol (RQ2)

Under the 70:10:20 partition used for training, 575 of 2,120 test images had a near-duplicate in training. That is 27.1%. Per class the figure runs from 5.0% in zinc up to 71.6% in healthy and 66.4% in sulphur, with the other seven all under 11% (Figure 4).

Simulating an 80:20 split with no validation set, over the same groups and five seeds, returned 32.16% (SD 1.42). The gap is expected — a larger training set catches more counterparts — but it is worth recording. Studies quoting contamination rates should say which split they assume. We use 27.1% below, since that is the split the experiments actually ran under.

5.3 Effect on accuracy (RQ3)

Table 4 and Figure 5 give six matched runs.

Table 4. Test accuracy of an identical MobileNetV3-Large under the two partitions across six matched runs.

Run	Partition A acc. (%)	Partition B acc. (%)	Δ acc. (pp)	Δ balanced (pp)	Test n
1	87.50	83.30	4.20	2.43	2,120
2	88.49	82.12	6.37	4.49	2,120
3	88.87	83.25	5.62	3.80	2,120
4	88.44	83.07	5.37	3.94	2,120
5	89.15	83.35	5.80	3.77	2,120
6	88.25	81.27	6.98	4.85	2,120
Mean	88.45	82.73	5.72	3.88	—
SD	0.57	0.85	0.94	0.82	—

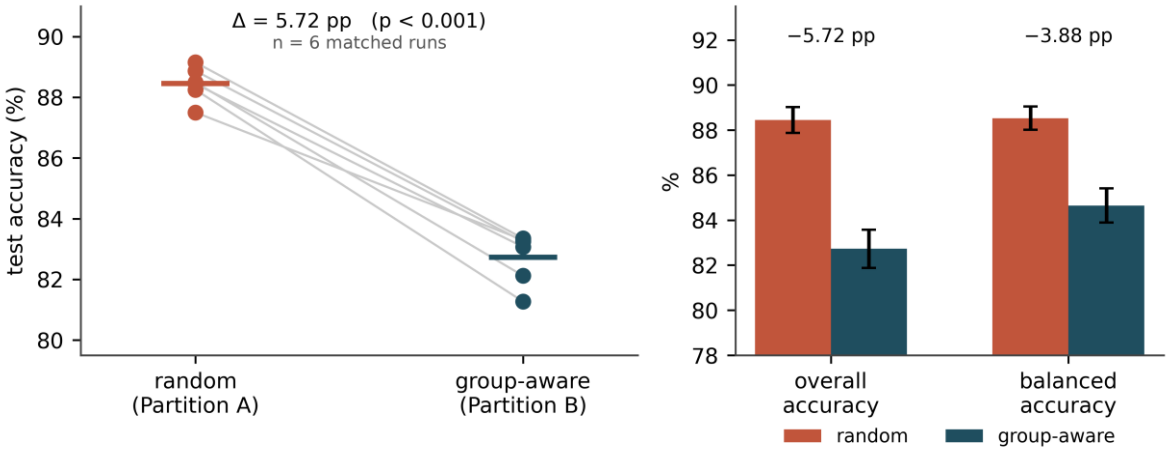

Figure 5. Left: accuracy for each of six runs, grey lines joining the conditions within a run, horizontal bars marking condition means. Right: mean overall and balanced accuracy with standard deviations.

The accuracy decreased by 5.72 points (SD 0.94), from 88.45% to 82.73%. Welch's test yields a 95% confidence interval of 4.8 to 6.7 points, $t = 13.7$ at $p < 0.001$, and a Cohen's d of around 7.9. All six runs move in the same

manner, and the spreads within conditions, 0.57 and 0.85, are far below the difference between them. This is not run-to-run noise.

From 88.53% to 84.65%, balanced accuracy decreased by 3.88 points. The first indication that the influence is not evenly distributed is the difference between the two measurements. The real location is shown in Sections 5.4 and 5.5.

5.4 Holding training data fixed: the contaminated and clean halves of one test set

Partition comparisons alter two things simultaneously. Although group-aware splitting eliminates contamination from the test set, it also modifies the training set since duplicate groups now arrive complete rather than dispersed. Both might result in a decrease in accuracy. In order to separate them, we trained a single model on Partition A and evaluated it on two separate segments of its own test set: the 1,545 images without a training counterpart and the 575 photos with one. Weights, hyperparameters, and training data are the same for each. Therefore, whatever varies is a characteristic of the test photos.

The model achieved 87.17% overall, 87.79% balanced, and 87.78% macro-F1 on the entire test set. Split by contamination, it scored 92.87% on the contaminated test images and 85.05% on the clean ones. That is a difference of 7.82 points, with two-proportion $z = 4.79$, $p = 1.7 \times 10^{-6}$, and a 95% bootstrap interval of 5.09 to 10.51. Taken alone, that settles the direction of the effect.

Balanced accuracy reverses it: 83.42% on the contaminated test images against 85.40% on the clean ones. Figure 6 illustrates why both claims are accurate. The contaminated group, which has 45% healthy and 34% sulphur—the exact classes the audit identified—compared to 7% and 6% in the clean subset, is nothing like a random sampling of the test set. With a recall of 94.2%, the easiest class in this model is healthy. Therefore, composition rather than memorization accounts for a significant portion of the 7.82-point differential.

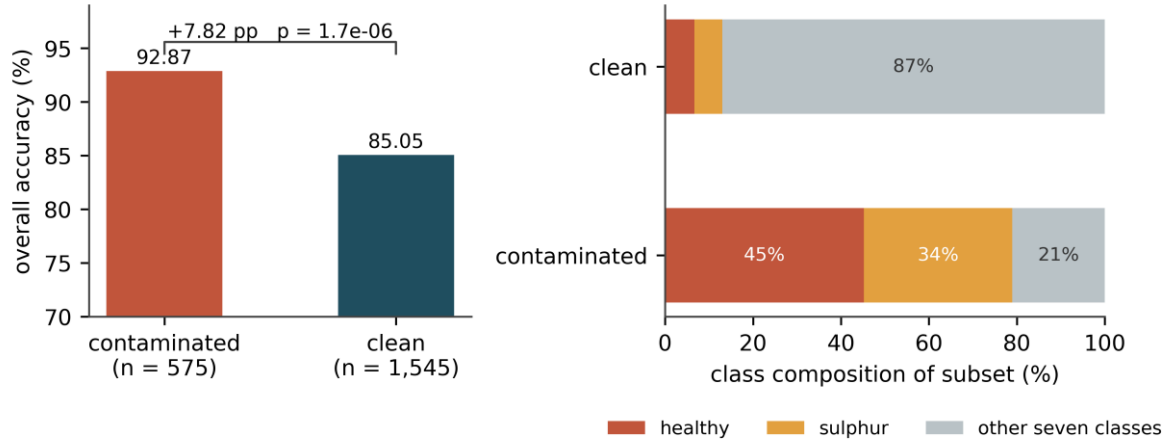

Figure 6. Left: accuracy of one model on the contaminated and clean parts of its own test set. Right: class composition of the two subsets. The contaminated subset is 79% drawn from the two entirely duplicated classes, which is why overall and balanced accuracy disagree about its difficulty.

The finding survives, but it needs stating more carefully. Contamination in this collection cannot be separated from class identity, for the simple reason that contamination is almost entirely a property of two classes. Section 5.5 asks the question at the level where an answer is available.

5.5 Per-class analysis

Table 5 and Figure 7 give recall under both partitions for every class.

Table 5. Per-class recall and F1 under both partitions, ordered by recall change. The two entirely duplicated classes occupy the top two rows. Δ recall is Partition A minus Partition B, so a positive value is a recall loss under group-aware splitting						
Class	**Dupl. (%)**	**Recall A (%)**	**Recall B (%)**	**F1 A (%)**	**F1 B (%)**	**Δ recall (pp)**
sulphur	100.0	86.64	72.34	84.19	74.45	+14.30
healthy	100.0	94.21	84.27	92.68	85.84	+9.95
calcium	11.3	73.52	70.00	76.17	72.50	+3.52
zinc	7.9	77.91	76.00	79.76	77.55	+1.91
boron	10.4	98.40	96.61	96.61	96.61	+1.79
magnesium	9.3	84.57	83.08	81.33	81.46	+1.49
iron	14.0	92.97	94.38	92.47	88.42	−1.41
manganese	14.1	97.62	100.00	97.62	95.91	−2.38
potassium	10.9	84.29	93.24	89.20	90.59	−8.95

Boron, confirmed: recall and F1 coincide at 96.61% under Partition B because precision and recall are equal there (both 96.61%), and F1 under Partition A equals the same figure by coincidence, precision 94.87% and recall 98.40% giving F1 96.61%. The three identical values are genuine, not a transcription error.

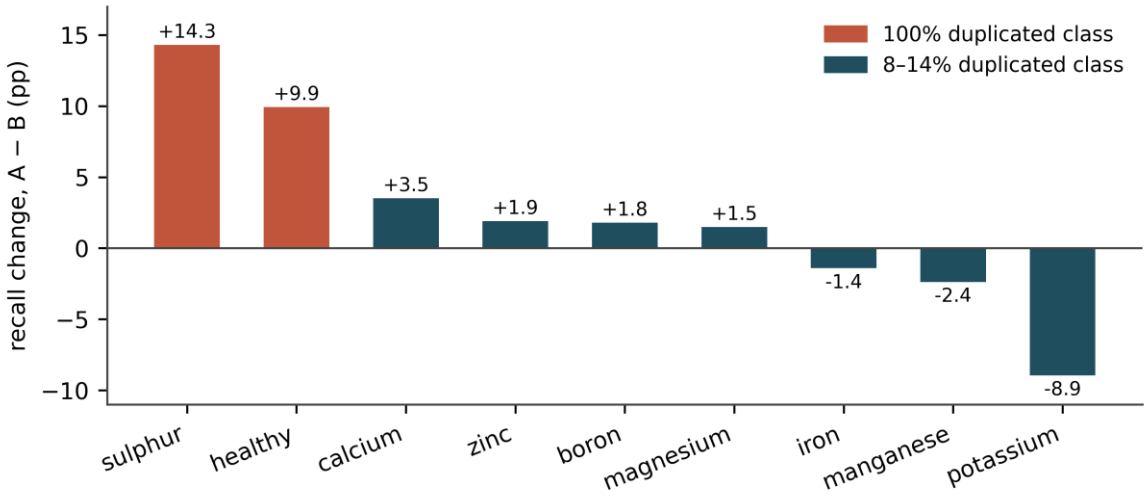

Figure 7. Change in per-class recall from Partition A to Partition B. The two classes composed wholly of duplicate groups lose 14.3 and 10.0 points. No other class loses more than 3.5, and three improve under group-aware splitting, potassium by 9.0 points.

The pattern is unambiguous. Keep duplicates out of the test set and sulphur loses 14.30 points of recall, healthy 9.95. None of the remaining seven loses more than 3.52, and three of them — iron, manganese, potassium — actually do better under group-aware splitting, potassium by 8.95 points. The two classes that collapse once the duplication is neutralized are precisely the ones that the audit identified as completely duplicated.

Compared to the total of 5.72 points, that is a far narrower causal assertion. It also explains the improvements: capacity is dispersed, classes like potassium benefit, and two simple, extensively replicated classes no longer flatter the model. Therefore, the total actually represents a weighted average of seven classes that hardly move or improve, one 14-point collapse, and one 10-point collapse.

6. Discussion

Six points of overall accuracy is not, on its own, a dramatic number. What makes it matter is the company it keeps. Gains of one to three points are frequently cited in papers in this field as proof that an architecture is superior. Rankings based on random splits of this dataset cannot be relied upon to withstand adjustment since an artifact of protocol worth almost six is greater than the signal those comparisons are designed to detect.

The more practical discovery is the structural one. The conclusion would have read, "39% of this collection is duplicated; fixing the split costs six points," if we had ended at the dataset level. It's true and practically useless for anyone trying to figure out what to do next. The view at the class level can be put into practice. Seven of the nine classes are sound; two are badly affected. The seven sound classes can be retained, while results for the sulphur and healthy classes should be treated as unreliable until that part of the benchmark is rebuilt. Contamination should be reported as a class-level breakdown rather than as a single headline percentage.

Consideration of the reversal in Section 5.4 is crucial, as other audits are likely to analyze it. Comparing subsets confounds contamination with class difficulty whenever the two are correlated, and the two metrics then disagree. The confounding variable will go unnoticed if only reporting overall accuracy—the result is a simple 7.82 point effect. Reporting balanced accuracy alone would make contamination look harmless. Only the two measures together, set beside the composition figure, give the full picture.

A plain remedy exists. Where a release carries separate raw and augmented folders, as this one does, augmentation belongs after partitioning rather than before, with the raw images alone defining the split. Costs nothing, removes the mechanism at the source. Where only augmented images are published, an audit along the lines of Algorithm 1 should come before any evaluation at all.

Kapoor and Narayanan recommend that papers document their partitioning explicitly [6], and these results bear that out. In this literature the procedure is usually given as a bare percentage, with nothing said about whether augmentation came before or after. Here that single omission is worth about six points of reported accuracy — and a great deal more than six in two of the nine classes.

7. Limitations

- One dataset, one crop. The mechanism is general, but the magnitude and the concentration in two classes may not be. Auditing several public collections would establish whether this pattern recurs.
- One architecture. Only MobileNetV3-Large was tested. Higher-capacity models memorise more readily and might show a larger gap; that is a direct and untested prediction.
- Threshold not varied. Detection used $\tau^{d} = 12$ and $\tau^{a} = 16$ throughout. The large group-count-to-size ratio and the 94% of groups spanning the raw and augmented folders argue against over-merging, and the largest group holds six images where over-merging produces hundreds, but a sweep across thresholds would place error bars on the 39.2% and the per-class figures. We regard this as the most important outstanding check.
- Detection recovers only mild transformations. Images rotated, cropped, or recoloured aggressively escape hashing, so duplication and contamination are lower bounds throughout.
- Verification was by inspection rather than by a quantified protocol. Contact sheets were examined and the groupings judged sound, but no sample size, second rater, or inter-rater agreement statistic was recorded. The visual evidence should therefore be read as corroboration of the automated grouping, not as a measured precision with confidence bounds.
- No specimen identifiers. The release records no leaf, plant, or acquisition-session identifier. Two photographs of different leaves from the same plant are statistically dependent, yet no appearance-based method can detect that dependence. Partition B removes appearance-level duplication; it does not establish full statistical independence between training and test. Multi-folder placement of a single leaf, which the data article

documents [17], is a related case: such images reach both partitions under a random split, but the mechanism is dataset design rather than error.

- Class and contamination are confounded. Because contamination is concentrated in two classes, no comparison of contaminated against clean images can fully separate the effect of duplication from the effect of class difficulty. Section 5.5 addresses this by comparing within classes, which is the strongest answer the data allow.
- Sensitivity to partition order. Two runs at the same seed returned contamination of 26.70% and 27.12%, because directory traversal order enters the construction of the image index. The variation is small, but it means a partition is exactly reproducible only when the file ordering is fixed.
- Exact reproduction unavailable. GPU convolution algorithm selection is nondeterministic under the default configuration, so results are reported as means over independent runs rather than as fixed seeded values.

8. Conclusion and Future Work

Two fifths of a widely used banana nutrient-deficiency dataset consists of near-duplicate images, but that figure describes the collection poorly. Two of the nine classes are duplicated in full, while the remaining seven average 11%. Under the training protocol, 27.1% of test images have a counterpart in training, ranging from 5% in zinc to 72% in healthy. Correcting the split costs 5.72 points of accuracy across six matched runs, significant at $p < 0.001$, and the per-class breakdown places that loss where the audit predicted: sulphur falls 14.3 points and healthy 10.0, no other class loses more than 3.5, and three classes improve, potassium by 9.0. Under one model, contaminated test images score 7.82 points higher than clean ones, though class composition accounts for much of that gap. That last point is a caution for anyone attempting the same comparison elsewhere.

Several lines of work follow directly.

1. Threshold sensitivity. Repeating the audit across a range of Hamming thresholds would bound the duplication and contamination estimates and settle how far the per-class concentration depends on the detector's settings. This is the first thing we intend to do.
2. A survey across datasets. Running Algorithm 1 over PlantVillage, PlantDoc and the coffee and rice deficiency collections would show whether concentrated duplication is peculiar to this release or common in the field. It needs no new method.
3. Architecture dependence. Repeating the comparison with residual networks, EfficientNets and compact vision transformers would test whether memorisation capacity governs the size of the effect.
4. Prevalence in the literature. Screening published work on these datasets for how many state their partitioning, and how many used random splits, would quantify how much of the record is affected.
5. Better detection. Substituting learned embeddings for perceptual hashing, scored against the groupings released here, would reveal how much duplication escapes appearance-based methods.
6. Cross-dataset generalisation with proper controls. Transfer to an independent field collection remains the open question, and requires a dataset containing verified healthy specimens so that a negative control is possible.
7. Corrected benchmark partitions. Publishing group-aware splits for the main public agricultural datasets would let the community report uncontaminated figures without each group repeating the audit.

Data and Code Availability

The image dataset analysed in this study is publicly available from Mendeley Data (doi: 10.17632/7vpdrbdkd4), version 1, downloaded on 13 September 2026, and is described in its data article [17]. The audit tool, the duplicate group listings, the group-aware partitions, and the training scripts used in this study have been deposited in a

permanent archive and are available at https://data.mendeley.com/datasets/7vpdrbdkd4/1. All experiments used publicly available data and required no funding, laboratory access, or specialised equipment.

Declaration of Generative AI Use

The authors used an AI assistant for language editing and manuscript preparation. All experimental design, data analysis, interpretation of results, and scientific conclusions are the authors' own. The authors reviewed and edited the content as needed and take full responsibility for the content of this article.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. No funding was received for this study. The research involved no human participants, no animal subjects, and no personal data.

Author Contributions (CRediT)

Yogesh: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Validation, Visualization, Writing – original draft, Writing – review & editing. Sudir Prakash: Methodology, Validation, Formal analysis, Writing – review & editing. Princy Randhawa: Methodology, Validation, Supervision, Writing – review & editing. All authors read and approved the final manuscript.

References

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Front. Plant Sci., vol. 7, art. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[2] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agric., vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[3] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," Biosyst. Eng., vol. 144, pp. 52–60, 2016, doi: 10.1016/j.biosystemseng.2016.01.017.

[4] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," Comput. Electron. Agric., vol. 153, pp. 46–53, 2018, doi: 10.1016/j.compag.2018.08.013.

[5] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: formulation, detection, and avoidance," ACM Trans. Knowl. Discov. Data, vol. 6, no. 4, pp. 1–21, 2012, doi: 10.1145/2382577.2382579.

[6] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 9, art. 100804, 2023, doi: 10.1016/j.patter.2023.100804.

[7] B. Barz and J. Denzler, "Do we train on test data? Purging CIFAR of near-duplicates," J. Imaging, vol. 6, no. 6, art. 41, 2020, doi: 10.3390/jimaging6060041.

[8] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?," in Proc. 36th Int. Conf. Machine Learning (ICML), 2019, pp. 5389–5400.

[9] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," in Proc. NeurIPS Datasets and Benchmarks Track, 2021.

[10] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv:1511.08060, 2015, doi: 10.48550/arXiv.1511.08060.

[11] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," Comput. Intell. Neurosci., vol. 2016, art. 3289801, 2016, doi: 10.1155/2016/3289801.

[12] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Comput. Electron. Agric., vol. 161, pp. 272–279, 2019, doi: 10.1016/j.compag.2018.03.032.

[13] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Nitrogen deficiency prediction of rice crop based on convolutional neural network," J. Ambient Intell. Humaniz. Comput., vol. 11, pp. 5703–5711, 2020, doi: 10.1007/s12652-020-01938-8.

[14] Yogesh, A. K. Dubey, R. R. Arora, and A. Rocha, "Multiclass classification of nutrients deficiency of apple using deep neural network," Neural Comput. Appl., vol. 34, no. 11, pp. 8411–8422, 2022, doi: 10.1007/s00521-020-05310-x.

[15] Yogesh, A. K. Dubey, R. Ratan, and A. Rocha, "Computer vision based analysis and detection of defects in fruits causes due to nutrients deficiency," Cluster Comput., vol. 23, no. 3, pp. 1817–1826, 2020, doi: 10.1007/s10586-019-03029-6.

[16] Yogesh, A. K. Dubey, and A. Rocha, "A non-invasive approach for calcium deficiency detection in pears using machine learning," Neural Comput. Appl., vol. 37, no. 11, pp. 7609–7618, 2025, doi: 10.1007/s00521-023-08444-w.

[17] Sunitha P, Uma B, Channakeshava S, and Suresh Babu C S, "A fully labelled image dataset of banana leaves deficient in nutrients," Data Brief, vol. 48, art. 109155, 2023, doi: 10.1016/j.dib.2023.109155.

[18] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," Biosyst. Eng., vol. 180, pp. 96–107, 2019, doi: 10.1016/j.biosystemseng.2019.02.002.

[19] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: a dataset for visual plant disease detection," in Proc. 7th ACM IKDD CoDS and 25th COMAD, 2020, pp. 249–253, doi: 10.1145/3371158.3371196.

[20] M. Roberts et al., "Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans," Nat. Mach. Intell., vol. 3, pp. 199–217, 2021, doi: 10.1038/s42256-021-00307-0.

[21] G. Varoquaux and V. Cheplygina, "Machine learning for medical imaging: methodological failures and recommendations for the future," npj Digit. Med., vol. 5, art. 48, 2022, doi: 10.1038/s41746-022-00592-y.

[22] S. Whalen, J. Schreiber, W. S. Noble, and K. S. Pollard, "Navigating the pitfalls of applying machine learning in genomics," Nat. Rev. Genet., vol. 23, pp. 169–181, 2022, doi: 10.1038/s41576-021-00434-9.

[23] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," M.Sc. thesis, Upper Austria Univ. Applied Sciences, Hagenberg, Austria, 2010.

[24] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proc. Int. Conf. Learning Representations (ICLR), 2019.

[27] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in Proc. 20th Int. Conf. Pattern Recognition (ICPR), 2010, pp. 3121–3124, doi: 10.1109/ICPR.2010.764.