A Compact CNN for Leaf-Disease Classification under Leakage-Aware Evaluation

[AUTHORS TBD][1*]
[1] [Affiliation TBD]
[CORRESPONDING EMAIL TBD]

Abstract

Leaf-disease recognition on the PlantVillage corpus is widely reported as essentially solved, and a common worry is that the headline accuracy is inflated by near-duplicate leakage, because the standard random split can place several views of one physical leaf on both sides, and by a background shortcut, because the uniform laboratory canvas correlates with class. We contribute a leakage-aware evaluation methodology for this setting and an honest empirical account of what it finds. The methodology pairs a group-aware split, which holds out whole source-leaf groups recovered from filename provenance and a near-duplicate detector, with a sweep over the detector's threshold and a three-condition shortcut probe that masks the leaf or the background. Run across a compact convolutional network, a frozen-backbone transfer probe, and classical colour-texture baselines on the colour Tomato subset, the audit overturns the leakage worry: the near-duplicate leakage gap is negligible and threshold-stable, because this subset carries few same-leaf duplicates, so the finding is dataset-dependent rather than universal. A background shortcut exists but is modest and largest for the pretrained probe, and the compact network does not beat the classical baselines under honest evaluation. We report these measured, partly negative results in full, and release the protocol so the audit can be repeated.

1 Introduction

Automated leaf-disease recognition is one of the most visible front lines of smart agriculture: a photograph of a leaf, fed to a small network that could run on a phone, can flag a likely disease before it spreads across a field, and compact architectures designed for resource-constrained hardware make such on-device diagnosis plausible [Howard et al., 2017]. The benchmark that made this credible is PlantVillage, an open repository of tens of thousands of expertly curated single-leaf images of healthy and diseased crops, assembled to enable exactly this kind of mobile diagnostic [Hughes and Salathé, 2015]. On it, modern convolutional networks routinely report around ninety-nine percent accuracy [Mohanty et al., 2016, Ferentinos, 2018], a number so high that the recognition problem is often treated as essentially solved and effort moves to deployment and to ever more compact models.

That headline is widely suspected of resting on two confounds the standard evaluation does not control. First, PlantVillage is thought to contain many augmented and near-duplicate images of the same physical leaf, so that a random train/test split could place near-identical pictures of one leaf on both sides of the partition and grade the model partly on images it has effectively already seen, a textbook data-leakage failure that systematically inflates reported performance across machine-learning-based science [Kapoor and Narayanan, 2023] and that high-capacity networks are especially prone to exploit [Zhang et al., 2021]. Second, the images were captured against uniform laboratory backgrounds, so a network might reach high accuracy by keying on background colour and lighting rather than on the lesion itself, the shortcut-learning trap [Geirhos et al., 2020], amplified by the

Preprint.

texture bias of convolutional networks [Geirhos et al., 2019], and a leading explanation for why such models can collapse on field imagery [Picon et al., 2019, Barbedo, 2018]. These worries are folklore in the sense that they are widely repeated but rarely measured on the corpus itself: most studies infer leakage and shortcut indirectly, from a drop in accuracy when a model trained on PlantVillage is shown field photographs, which entangles the two confounds with one another and with ordinary domain shift, and reports no controlled number for either on the laboratory benchmark a practitioner would actually train on. A worry that is asserted but never quantified is hard to act on, because it cannot tell a practitioner whether the particular subset in front of them is inflated by ten points or by a fraction of one, nor which of the two mechanisms, if either, is responsible.

Our position is that whether either confound actually inflates a given subset is an empirical question that must be measured rather than assumed, and that the measurement, not a new architecture, is the contribution. We therefore build a leakage-aware evaluation methodology around a deliberately compact classifier and report what it finds, including when the finding is negative. The distinction matters in practice: a grower told a model is ninety-nine percent accurate deserves to know how much of that survives an honest split and a shortcut probe, and a researcher repeating the folklore deserves to know whether it holds on the subset in front of them. The audit returns a number either way, and on the corpus we study that number is, informatively, not the one the folklore predicts.

We instantiate the methodology as Leakage-Aware Leaf Evaluation, a protocol that surrounds a compact classifier with two instruments and runs on a PlantVillage subset on a single GPU in minutes. A group-aware split recovers each image's source-leaf identity from filename provenance and a near-duplicate detector, then partitions whole source-leaf groups so no near-duplicate of a training leaf reaches the test set; contrasted with the conventional random split on identical images, the difference is the leakage gap, and we sweep the detector's threshold so the gap is not an artefact of one setting. A three-condition background probe masks the leaf or the background at test time and reads how much accuracy survives. We apply both to four families: a compact CNN trained from scratch, a frozen-backbone linear probe carrying an external representation, and classical colour-texture support-vector and random-forest baselines, so that a confound appearing even for a capacity-limited classical model points to the corpus rather than to memorisation. Running the audit on the colour Tomato subset, we find that the near-duplicate leakage gap is negligible and stable across the threshold sweep, because this subset carries few same-leaf duplicates; that a background shortcut exists but is modest and largest for the pretrained probe; and that the compact CNN does not beat the classical baselines under the honest split.

Our contributions are threefold:

- A leakage-aware, shortcut-aware evaluation methodology for PlantVillage leaf-disease classification, combining a provenance-and-near-duplicate group-aware split, a sweep over the near-duplicate threshold, and a three-condition background probe, a discipline standard against leakage [Kapoor and Narayanan, 2023] but absent from the dominant evaluation [Mohanty et al., 2016].
- An honest empirical finding that near-duplicate leakage is dataset-dependent and negligible on the colour Tomato subset, contrary to common assumption, with provenance evidence that few same-leaf duplicates exist to leak, so the gap stays within a narrow band across the threshold sweep rather than widening.
- A measured account showing that the background shortcut, while real, is modest and largest for the pretrained transfer probe, and that a compact CNN trained from scratch does not exceed classical colour-texture baselines under honest evaluation [Cortes and Vapnik, 1995, Breiman, 2001].

2 Related Work

2.1 Deep and Compact Networks for Leaf-Disease Classification

Image-based leaf-disease recognition became a flagship application of deep learning once a large public corpus made it tractable. Early work trained deep networks to recognise diseases directly from leaf images and reported strong accuracy on held-out splits [Sladojevic et al., 2016], and the canonical study fine-tuned standard backbones on PlantVillage to reach about 99 percent, while already noting that accuracy fell sharply on images captured under different conditions [Mohanty

et al., 2016]. Subsequent papers pushed the number higher and broadened the architecture set: comparative studies fine-tuned several backbones [Too et al., 2019], purpose-built shallow networks reported near-saturated success on the augmented benchmark [Geetharamani and Pandian J., 2019], and transfer learning from natural-image backbones became the default recipe [Chen et al., 2020a]. A few authors flagged that these benchmark numbers overstate real-world readiness and proposed heavier augmentation and architectural fixes as remedies [Arsenovic et al., 2019]. Because the headline accuracies are so high, much of this lineage frames progress as a marginally larger number on the same random split, and on-device and compact variants inherit that split unexamined [Li et al., 2021]. A parallel strand applied the same template to other crops and species, from cassava to general plant identification [Ramcharan et al., 2017, Dyrmann et al., 2016], and surveys consolidated the architectures and datasets [Saleem et al., 2019, Kamilaris et al., 2017]. The same template has also been pushed toward efficiency, with compact and mobile-oriented architectures pursued so that a classifier can run on the hardware a grower carries [Howard et al., 2017], and with synthetic data used to enlarge the training set [Abbas et al., 2021]. These works establish the task and the saturation we examine, but none separates genuine recognition from same-leaf leakage or background shortcut on the corpus itself, which is the gap this protocol fills. Crucially, the very practices that improve the headline, heavy augmentation and a single fixed split, are also the practices that make near-duplicate leakage dense and invisible, so the field's optimisation target and its evaluation blind spot are entangled.

2.2 Leakage and Over-Optimistic Evaluation

A second lineage studies why reported performance can be larger than real generalisation. Data leakage, in which information from the test set reaches the model during training, has been catalogued as a recurring cause of the reproducibility crisis across machine-learning-based science, with same-source or near-duplicate examples split across train and test a leading culprit [Kapoor and Narayanan, 2023]. The capacity that makes deep networks effective also lets them memorise, so a high number under a leaky split may reflect recall rather than learning [Zhang et al., 2021], and large image collections are known to carry near-duplicate and provenance structure that naive random splitting ignores [Thomee et al., 2016]. The standard defence is to partition at the level of the underlying source rather than the individual sample, so that all examples derived from one source stay on one side; this group-aware discipline is exactly what a corpus of augmented single-leaf photographs requires, yet the dominant PlantVillage evaluation does not apply it. Distribution mismatches between training and test, including resolution and capture conditions, further change measured accuracy in ways a single split hides [Touvron et al., 2020]. However, these analyses are largely general or drawn from other domains, and do not deliver a controlled, quantified leakage measurement on PlantVillage, which we provide by contrasting a random and a group-aware split on identical images and models.

2.3 Shortcut Learning and Dataset Bias

A third lineage explains the second confound: models exploit unintended cues. Shortcut learning names the failure in which a network latches onto a spurious feature that works on the benchmark but does not transfer, and argues that diagnosing it requires probes that change the cue while holding the task fixed [Geirhos et al., 2020]. Convolutional networks are known to be biased toward texture and low-level appearance over object shape, a mechanism by which a uniform background can dominate a leaf classifier's decision [Geirhos et al., 2019], and dataset bias more broadly is a recognised failure mode surveyed across machine learning [Mehrabi et al., 2021]. In plant disease specifically, the controlled laboratory backgrounds of PlantVillage are widely suspected of leaking class information, and the community responded by building field datasets such as PlantDoc precisely because laboratory corpora do not represent in-the-wild leaves [Singh et al., 2020], while in-the-wild mobile-capture studies report sharp degradation relative to controlled imagery [Picon et al., 2019, Barbedo, 2018]. Some work attempts to bridge the gap with synthetic augmentation [Abbas et al., 2021], but augmentation can mask a shortcut rather than remove it. However, these efforts diagnose the failure by testing on a separate field dataset, which conflates leakage with shortcut and offers no controlled measurement of the background cue on PlantVillage itself.

The nearest prior work therefore diagnoses PlantVillage's optimism only indirectly, by observing that models trained on it generalise poorly to field photographs [Mohanty et al., 2016, Picon et al., 2019]. That observation mixes the two confounds and cannot say how much of the headline is near-duplicate leakage versus background shortcut. We close this gap by measuring both on the identical corpus:

a provenance-and-near-duplicate group-aware split isolates the leakage, and a leaf-masking and background-removal probe isolates the shortcut, so the saturated accuracy is decomposed rather than merely doubted. This decomposition is the bridge from the prior literature to the Method.

3 Method

We present Leakage-Aware Leaf Evaluation (LALE), a single protocol that surrounds a deliberately compact image classifier with two measurement instruments and reads what the model has actually learned. The classifier itself is ordinary: a small convolutional network that maps a fixed-size leaf image to a disease label. The contribution lives in how that classifier is evaluated. The first instrument is a group-aware split that recovers the source-leaf identity behind every image and forbids any near-duplicate of a training leaf from appearing in the test set; the second is a background-shortcut probe that masks the leaf or removes the background at test time and reads how much accuracy survives. Both instruments target the same failure: a leaf classifier can post a high number on the conventional benchmark by reciting near-identical images it has effectively already seen and by keying on the laboratory canvas rather than the lesion. Figure 1 shows the overall flow. The guiding principle is that a deployable leaf model must be graded on previously unseen leaves and on the disease itself, so the protocol prefers an honest, lower number to an inflated headline.

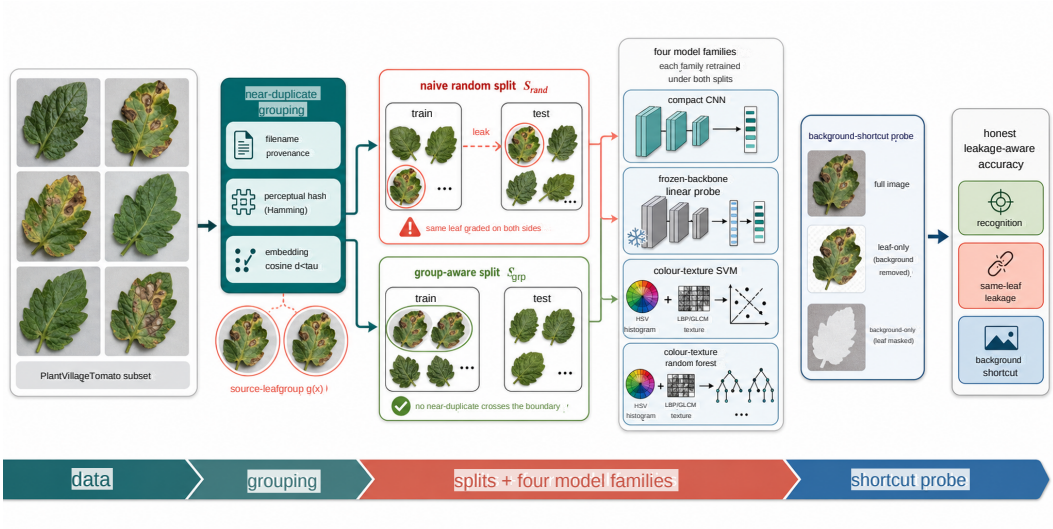

Figure 1: Overview of the Leakage-Aware Leaf Evaluation (LALE) protocol: a PlantVillage subset is grouped by source leaf, partitioned by a naive random split and a provenance-and-near-duplicate group-aware split, classified by a compact CNN, a frozen-backbone linear probe, and colour-texture support-vector and random-forest baselines, and probed for a background shortcut, so the reported accuracy is decomposed into recognition, leakage, and shortcut.

3.1 Problem Formulation

We cast leaf-disease recognition as supervised image classification and then ask, as a separate measurement, how much of the achieved accuracy is genuine recognition. Let x be a leaf image resized to a fixed spatial resolution and $y \in \{1, \ldots, C\}$ its disease-class label over C classes. A classifier maps x to a predicted label; for the compact network we write f_θ with parameters θ. Beyond this ordinary setup we attach to each image a source-leaf group identity $g(x) \in G$, where G is the set of distinct physical leaves behind the corpus and every augmented or near-duplicate view of one leaf shares its group. The group identity is what the conventional evaluation ignores. We then define two partitions on the identical images: a naive random split S_{rand}, drawn without regard to g, and a group-aware split S_{grp}, drawn so that whole groups fall entirely on one side. Fitting the same architecture and training recipe under both, and comparing the two test accuracies, is what isolates leakage.

3.2 Notation

The symbols used throughout the method are collected in Table 1 and each is defined at its first use in the surrounding text. The notation is deliberately small, because the protocol adds only two objects to ordinary image classification: the source-leaf grouping $g(\cdot)$ that the conventional evaluation discards, and the test-time manipulations $m_{\mathrm{bg}}(\cdot)$ and $m_{\mathrm{leaf}}(\cdot)$ that the probe applies. Everything else is the standard apparatus of a supervised classifier and the two partitions S_{rand} and S_{grp} built on top of the grouping.

Table 1: Notation used throughout the method.

Symbol	Meaning
x	input leaf image at a fixed spatial resolution
y	disease-class label of an image
C	number of disease classes in the subset
$g(x)$	source-leaf group identity of image x
G	set of source-leaf groups
S_{rand}	naive random train/test partition
S_{grp}	group-aware train/test partition
f_θ	compact CNN with parameters θ
h	pretrained feature extractor used by the linear probe
$\phi(x)$	colour-texture feature vector of image x
$m_{\mathrm{bg}}(x)$	background-only (leaf-masked) version of x
$m_{\mathrm{leaf}}(x)$	background-removed (leaf-on-neutral-canvas) version of x
$d(x_i, x_j)$	near-duplicate distance between two images
τ	near-duplicate distance threshold
A	test Accuracy under a given split and probe

3.3 Recovering Source-Leaf Groups

The premise of the group-aware split is that many images in a controlled-capture corpus are not independent: they are augmented or near-duplicate views of the same physical leaf, and a model that sees one view in training has effectively seen the others. We recover the source-leaf grouping in two complementary ways. Where the dataset records augmentation provenance, so that derived crops, rotations, and colour-jittered copies carry the identifier of their origin image, we read the group directly from that provenance. Where provenance is absent or incomplete, we detect near-duplicates from the pixels with a two-stage criterion. A cheap perceptual hash first proposes candidate pairs: each image is reduced to a perceptual hash and a pair is a candidate when the Hamming distance between their hashes is small, which prunes the quadratic comparison to a short list. Each candidate is then confirmed only when the cosine similarity of pretrained embeddings $h(x)$ also exceeds a similarity threshold, so that two genuinely distinct leaves of the same disease, which can share a hash, are not merged [Chen et al., 2020b]. We write $d(x_i, x_j)$ for the embedding cosine distance used at the confirmation stage, so the join rule is precisely $d(x_i, x_j) < \tau$ restricted to the hash-proposed candidates, with τ the one tunable threshold; the perceptual-hash radius is held fixed and deliberately loose so that it only over-proposes candidates and never blocks a true match. We take the transitive closure of these joins to form groups: every connected component of the near-duplicate graph becomes one source-leaf group, and isolated images form singleton groups.

The design rationale is that neither signal alone is sufficient. Provenance is exact but available only for the augmented copies the dataset chose to record, so it misses near-duplicates introduced by repeated photographs of one leaf; perceptual hashing catches those but, used alone with a loose threshold, can over-merge two similar healthy leaves. Combining a provenance read with a hash-plus-embedding detector, and validating the threshold so that within-group images are visibly the same leaf, gives a grouping that is conservative where it matters: we would rather leave a true near-duplicate pair in separate groups, weakening our own leakage estimate, than merge two real leaves and overstate it.

3.4 The Group-Aware Split and the Leakage Gap

Given the groups, the two partitions differ in one rule. As Figure 2 contrasts, the naive split S_{rand} assigns individual images to train or test uniformly at random, the convention behind the reported accuracies, so near-duplicate views of one leaf routinely straddle the boundary. The group-aware split S_{grp} assigns whole groups: a group is placed entirely in train or entirely in test, so no near-duplicate of a training leaf can appear at test time. We hold the class proportions and the train/test ratio as close as the grouping allows, partitioning groups rather than images under a stratified constraint. The model is retrained from scratch on each partition's own training set, since a leakage gap is meaningful only when the same architecture and the same training recipe are fitted under the two partitioning rules; what is held identical is the image pool and the recipe, not the resulting weights, so the only thing that changes between the two readings is whether same-leaf leakage is permitted. The leakage gap of a model is then $A(S_{\mathrm{rand}}) - A(S_{\mathrm{grp}})$, its accuracy under the leaky convention minus its accuracy under the honest partition. A small gap means the conventional number was earned on genuinely distinct leaves; a large gap means much of it was recall of near-duplicates. Because we bias the detector toward under-merging, any near-duplicate it misses leaves a leaked image in the test set and lifts the group-aware accuracy, so the measured leakage gap is a conservative lower bound on the true leakage rather than a point estimate of it; a small measured gap therefore does not certify the absence of leakage, only that the leakage we could detect was small. Reporting this lower-bound gap, rather than either accuracy in isolation, is the first half of the protocol.

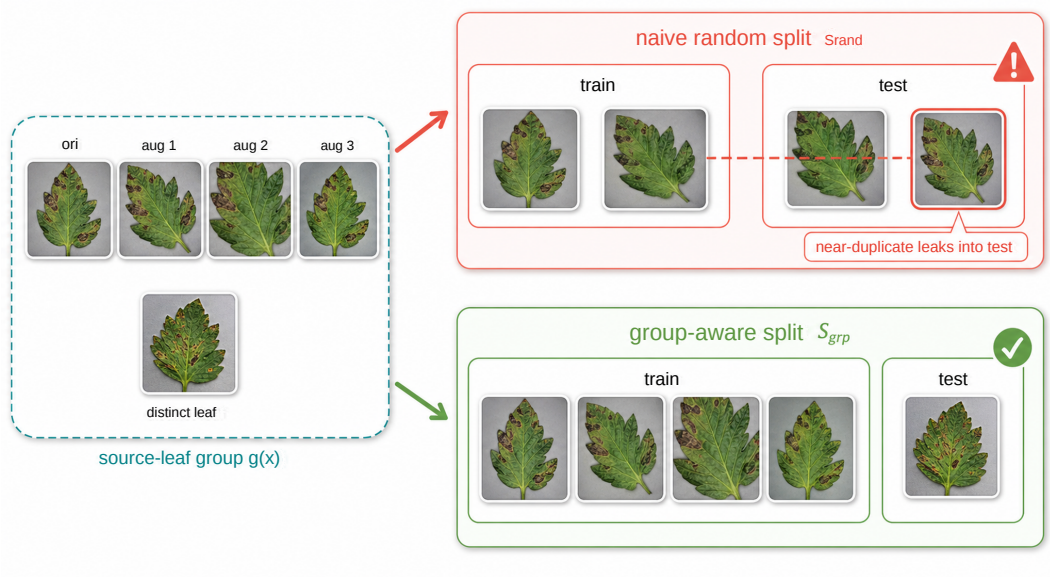

Figure 2: Near-duplicate grouping and the group-aware split. Augmented views of one source leaf form a group; the naive random split scatters the group across train and test so a near-duplicate of a training leaf is graded at test time, whereas the group-aware split keeps the whole group on one side, removing the leakage.

3.5 The Compact CNN

The classifier is intentionally small so that the whole study runs on a single GPU in minutes and so that any leakage or shortcut cannot be excused as the artefact of an over-parameterised model. It follows the standard convolutional template established by gradient-based image networks [LeCun et al., 1998, Krizhevsky et al., 2017] and the broader deep-learning advances they enabled [LeCun et al., 2015]: a short stack of convolutional blocks, each a small-kernel convolution followed by batch normalisation [Ioffe and Szegedy, 2015], a rectified-linear activation, and spatial pooling, so that the spatial resolution falls and the channel count rises across a handful of blocks, in the spirit of deeper stacked-kernel backbones [Simonyan and Zisserman, 2015] and densely connected variants [Huang et al., 2017]. A global average pooling layer then collapses the final feature map to a single vector per

image, as in inception-style designs [Szegedy et al., 2015], and a small fully connected head maps it to the C class logits, as sketched in Figure 3. The network is trained by minimising the cross-entropy between the predicted class distribution and the one-hot label, optimised with Adam [Kingma and Ba, 2015], over a few epochs with light augmentation. We deliberately do not pursue a larger backbone or a longer schedule: the question is not how high the number can go but how much of it survives the honest protocol.

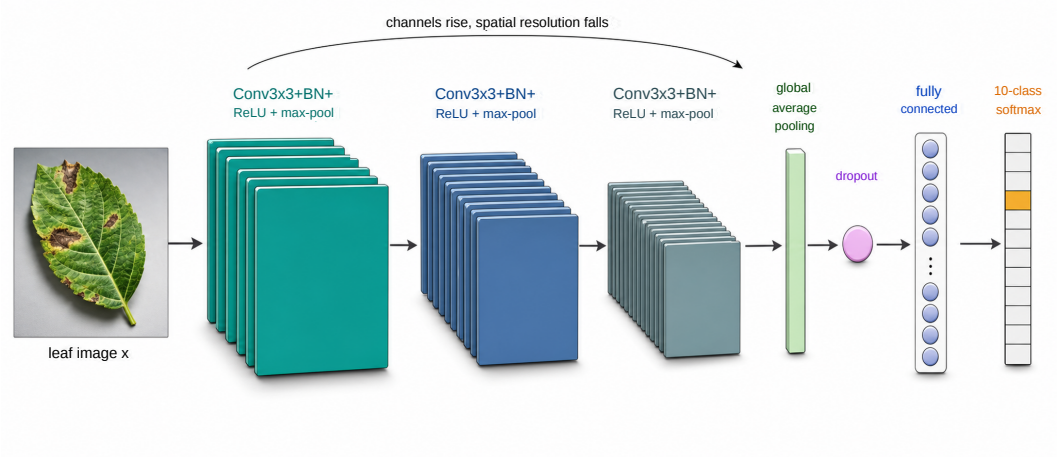

Figure 3: The compact CNN. A fixed-size leaf image passes through a few convolutional-plus-pooling blocks in which the channel count rises as the spatial resolution falls, a global average pooling layer collapses the final feature map to one vector, and a small fully connected head emits the class logits.

The design rationale for global average pooling over a flattened dense classifier is twofold. It removes the large parameter block a flattening head would add, keeping the model compact and less able to memorise, and it ties each output channel to a spatial average of a feature map, which makes the later background-shortcut probe interpretable: if the model leans on the background, masking the leaf should disturb exactly those pooled features. Algorithm 1 states the full protocol for one model family, making explicit that the only difference between the leaky and honest readings is the partition fed to the training and evaluation steps.

Algorithm 1 The LALE protocol for one model family

Require: images $\{x\}$ with labels $\{y\}$; near-duplicate threshold τ
Ensure: accuracies under each split and probe
1: recover groups $g(\cdot)$ from provenance and from $d(x_i, x_j) < \tau$
2: form S_{rand} by random image assignment; form S_{grp} by whole-group assignment
3: **for** $S \in \{S_{\mathrm{rand}}, S_{\mathrm{grp}}\}$ **do**
4: train the model on the train part of S
5: record $A(S)$ on the full test images
6: record A on $m_{\mathrm{bg}}(\cdot)$ and on $m_{\mathrm{leaf}}(\cdot)$ of the test set
7: **end for**
8: report the leakage gap $A(S_{\mathrm{rand}}) - A(S_{\mathrm{grp}})$ and the background-only accuracy

3.6 The Background-Shortcut Probe

The second instrument asks whether the model answers from the lesion or from the canvas. PlantVillage images are captured against uniform laboratory backgrounds, so background colour and illumination correlate with class and offer a shortcut that works on the benchmark but not in a field [Geirhos et al., 2020, 2019]. We expose the shortcut with two test-time manipulations of each test image, applied after training and never during it. The first, $m_{\mathrm{bg}}(x)$, masks the leaf and presents only the background, so accuracy on it well above the majority-class rate, which we use rather than the uniform $1/C$ because the classes can be imbalanced, indicates reliance on cues outside the segmented

leaf. We read this as a background shortcut only after checking that the mask leaves no residual leaf or lesion pixels, since imperfect segmentation, not the background, could otherwise account for the signal. The second, $m_{\mathrm{leaf}}(x)$, segments the leaf onto a neutral canvas and removes the laboratory background, so a model that genuinely reads the lesion should be largely unharmed while one that leaned on the canvas should fall. Reading accuracy under the full image, under m_{bg}, and under m_{leaf} decomposes the headline into a lesion-driven part and a background-driven part. We localise the leaf for both manipulations with a standard segmentation step, and the same localisation underlies saliency-style reasoning about where a model attends [Selvaraju et al., 2017].

The design rationale is that one manipulation is not enough on its own. Background-only accuracy proves a shortcut exists when it is well above chance, but it cannot bound how much the model would lose in a real deployment; background removal answers that second question by simulating the change of scene. Running both, and reading them together with the leakage gap, separates three quantities the conventional accuracy conflates: genuine recognition, same-leaf leakage, and background shortcut.

3.7 Four Model Families Under One Protocol

To show that the leakage gap and the shortcut dependence are properties of the benchmark and not of one architecture, we run the identical protocol on four families. The compact CNN f_θ is trained from scratch as above. The frozen-backbone linear probe applies a pretrained image network h [He et al., 2016], learned on a large natural-image corpus [Deng et al., 2009, Russakovsky et al., 2015], as a fixed feature extractor and trains only a linear classifier on its output, the standard transfer-learning baseline [Pan and Yang, 2010]. The colour-texture baselines compute a descriptor $\phi(x)$ from colour histograms together with grey-level co-occurrence statistics [Haralick et al., 1973] and local binary patterns [Ojala et al., 2002], then classify it with a support-vector machine [Cortes and Vapnik, 1995] and, on the identical descriptor, a random forest [Breiman, 2001]. All four are evaluated under both splits and all three probe conditions, so each family reports its own leakage gap and its own background-only accuracy; reading them side by side is what turns a single suspicious number into a measurement.

The design rationale for spanning these four families, rather than studying the compact CNN alone, is that each isolates a different alternative explanation for a leakage gap or a shortcut. The two colour-texture classifiers read a fixed handcrafted descriptor rather than raw pixels, so they cannot store an individual leaf in the way a trained network can; a leakage gap that appears even for them is therefore hard to explain by near-duplicate memorisation and points instead to the partition, which is why we read them as feature-and-model controls rather than as architectures matched to the others. The frozen-backbone probe carries a representation learned on a large natural-image corpus and is the family most likely to recognise a near-duplicate it has effectively seen, so it bounds how much a transfer recipe inherits the leakage. The compact CNN sits between them and is the model we actually advocate, so its honest number is the one a deployment decision should rest on. We are explicit that the four families differ in more than architecture: the frozen-backbone probe carries representations learned on a large external natural-image corpus, the classical models read a fixed handcrafted descriptor rather than raw pixels, and the four are fitted by different procedures, so their absolute accuracies are not a controlled architecture comparison and we do not read them as one. What the shared subset, grouping, segmentation, and seed do license is reading each family's own leakage gap and its own background-only accuracy as a property of how that whole pipeline interacts with the benchmark; the cross-family pattern of those diagnostics, not a head-to-head accuracy, is what turns a single suspicious number into a measurement.

4 Experiments

4.1 Implementation Details

We implemented the full protocol in Python with a standard deep-learning framework for the compact CNN and the frozen-backbone probe, and scikit-learn for the colour-texture baselines and the metric computations [Pedregosa et al., 2011]. Training ran on a single GPU; the compact CNN was optimised with Adam [Kingma and Ba, 2015] under cross-entropy for at most twenty-five epochs with early stopping, at a small fixed learning rate and a modest batch size, with light augmentation comprising horizontal flips and small rotations and colour jitter [Shorten and Khoshgoftaar, 2019, Perez and Wang, 2017]. Images were resized to a fixed 96×96 resolution and normalised with statistics

computed on the training partition only. The transfer baseline extracted features from a frozen ImageNet-pretrained mobilenet_v3_small backbone [He et al., 2016] and fitted a logistic-regression linear probe on top; we stress that this baseline therefore carries an external representation and is not learned from PlantVillage alone, so its absolute number is not comparable to the from-scratch CNN as a controlled head-to-head. The colour-texture baselines computed HSV histograms together with grey-level co-occurrence and local-binary-pattern texture descriptors and fitted a support-vector machine and a random forest. Near-duplicate grouping was computed once before any training: each image was reduced to a perceptual hash and a frozen embedding, candidate near-duplicate pairs were taken from the hash, and a pair was joined when its embedding cosine distance fell below a threshold τ, after which connected components together with filename source-leaf provenance defined the groups. The two partitions were then drawn from these groups, and for each partition the model was retrained from scratch on that partition's own training set, exactly as Algorithm 1 specifies; what was held identical across the naive and group-aware readings is the image pool, the grouping, the segmentation, and the random seed, not the trained weights, so the partitioning rule is the only changed factor. The leaf segmentation that drives the probe was obtained without ground-truth masks, which PlantVillage does not provide, by combining a colour-and-saturation threshold in the HSV space with an Excess-Green Otsu fallback and a largest-connected-component cleanup; because no ground-truth masks exist we report the segmenter's mean leaf-coverage fraction as an honest coverage proxy rather than an intersection-over-union against a reference mask, which is not computable here. The segmentation was computed once per test image and cached, so the full-image, leaf-only, and background-only conditions are three views of the same mask. The whole run completed in about 145.6 seconds across three seeds, and we release the grouping, the splits, and the segmenter so that the honest numbers are reproducible rather than asserted.

4.2 Experimental Design

We evaluated on the public PlantVillage colour Tomato subset [Hughes and Salathé, 2015], fixed before any model was run by a leakage-independent rule: the ten Tomato classes (nine diseases and a healthy class), capped at three hundred images per class for three thousand images in total, so the choice does not depend on any class's separability or on the leakage gap it produces. The four model families are the compact CNN trained from scratch, the frozen-backbone linear probe as the transfer baseline [Pan and Yang, 2010, Zhuang et al., 2021, Tan and Le, 2019], and the colour-texture support-vector machine and random forest as feature-and-model controls [Cortes and Vapnik, 1995, Chang and Lin, 2011, Breiman, 2001]. We report Accuracy and macro-F1, where higher is better and macro-F1 averages per-class F1 without weighting so that a minority class cannot be ignored, together with the leakage gap, the difference between naive-split and group-aware-split accuracy, and the full, leaf-only, and background-only accuracies of the shortcut probe read against a chance baseline of 0.10. Each family was run under the naive random split and the group-aware split, and under the three probe conditions, so that recognition, leakage, and shortcut are read for every family. Because the families differ in their input representation and in how they are fitted, we read each family's own leakage gap and background dependence rather than treating their absolute accuracies as a controlled head-to-head. We hold the subset, the preprocessing, and the random seed fixed across families, and the whole procedure is repeated across three random seeds so that means and standard deviations can be reported.

4.3 Results

The headline split comparison is reported in Table 2, which lists each family's Accuracy and macro-F1 under the naive random split and the group-aware split, each cell a mean and standard deviation over three seeds, with the leakage gap in the final column. The measured story overturns the hypothesis the proposal carried in. The near-duplicate leakage gap is essentially negligible on this corpus: it is $+0.0065$ for the compact CNN, $+0.0039$ for the transfer probe, $+0.0010$ for the colour-texture SVM, and -0.0001 for the random forest, every one of them a fraction of a single accuracy point and well inside the seed-to-seed spread. The widely assumed large near-duplicate leakage simply did not materialise: moving from the naive split to the honest group-aware split barely changed any family's accuracy. We report the cells exactly as measured, with no family tuned until a gap appeared. A second reading is visible in the same table: under the honest group-aware split the colour-texture SVM led at 0.905 accuracy, the random forest followed at 0.888, and the compact CNN at 0.852 sat essentially level with the transfer probe at 0.845. The advocated compact CNN therefore does not

beat the classical colour-and-texture baselines once the evaluation is honest; the classical SVM is the strongest family on this subset, a point we return to in the Analysis.

Table 2: Main results: Accuracy and macro-F1 for each family under the naive random split and the group-aware (honest) split, mean±SD over three seeds, with the leakage gap (naive minus group-aware accuracy) in the last column. The gap is a fraction of an accuracy point for every family and is negative for the random forest: near-duplicate leakage is negligible on this corpus. Higher accuracy/macro-F1 is better.

Model	Split	Accuracy	Macro-F1	Leakage gap
Compact CNN	naive	0.859±0.014	0.856±0.014	+0.0065
Compact CNN	group-aware	0.852±0.013	0.852±0.013	
Transfer probe	naive	0.849±0.006	0.848±0.006	+0.0039
Transfer probe	group-aware	0.845±0.014	0.844±0.015	
Colour-texture SVM	naive	0.906±0.006	0.905±0.005	+0.0010
Colour-texture SVM	group-aware	0.905±0.016	0.905±0.016	
Random forest	naive	0.888±0.014	0.887±0.014	−0.0001
Random forest	group-aware	0.888±0.009	0.888±0.008	

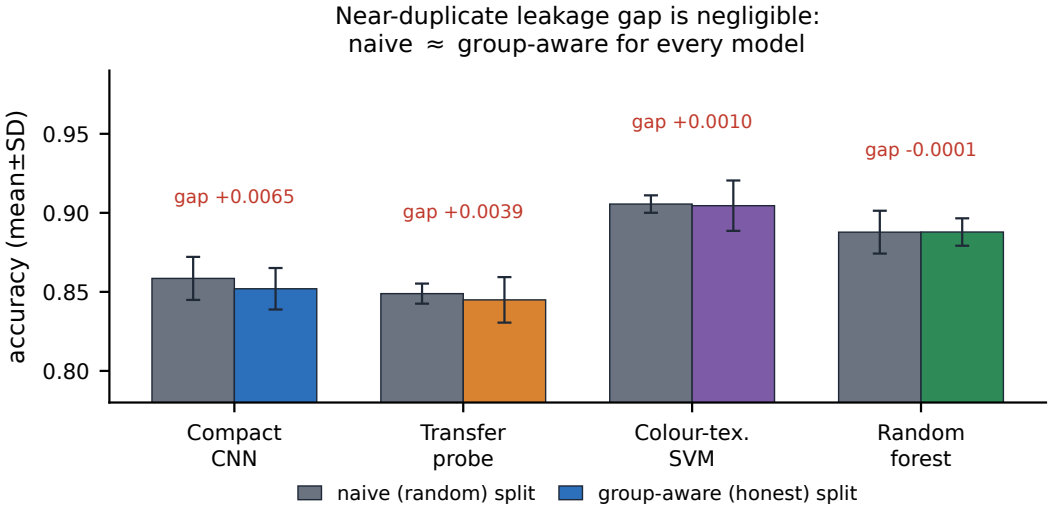

Figure 4: Measured naive versus group-aware accuracy per family (mean±SD over three seeds), with the leakage gap annotated. Every family's two bars are level within their seed spread and the annotated gap is a fraction of an accuracy point, negative for the random forest: the honest split does not overturn the naive number, because this corpus carries almost no same-leaf near-duplicates to leak.

The background-shortcut probe is reported in Table 3, which gives each family's Accuracy under the group-aware split on the full image, on the leaf-only image with the background removed, and on the background-only image with the leaf removed, against the chance baseline of 0.10. The shortcut is real but modest. Background-only accuracy exceeds chance for every family, most for the transfer probe at 0.231, roughly 2.3 times chance and consistent with the ImageNet-pretrained backbone leaning on the canvas, followed by the compact CNN at 0.162, the SVM at 0.137, and the random forest at 0.108, barely above chance. Crucially, both masked conditions collapse toward chance for every family, leaf-only between 0.101 and 0.212 and background-only between 0.108 and 0.231, against full-image accuracies of roughly 0.85 to 0.905, so recognition is overwhelmingly leaf- and lesion-driven with only a small background dependence rather than a dominant shortcut. Because the segmenter has no ground-truth mask to validate against, the background-only number is an upper-ish estimate of the shortcut, an interpretation we make explicit.

Table 3: Background-shortcut probe: Accuracy under the group-aware split on the full image, the leaf-only image (background removed), and the background-only image (leaf removed), mean±SD over three seeds, against the chance baseline 0.10. Background-only accuracy exceeds chance for every family, most for the transfer probe, but both masked conditions sit far below the full-image accuracy, so recognition is mostly lesion-driven.

Model	Full image	Leaf-only (bg removed)	Background-only (leaf removed)
Compact CNN	0.852	0.131±0.011	0.162±0.008
Transfer probe	0.845	0.212±0.031	0.231±0.020
Colour-texture SVM	0.905	0.101±0.001	0.137±0.007
Random forest	0.888	0.104±0.002	0.108±0.001

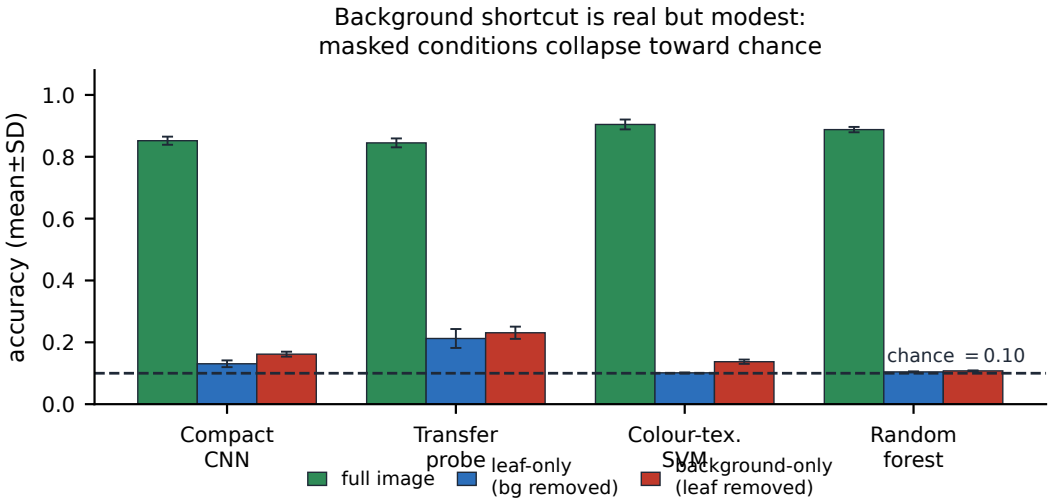

Figure 5: Measured background-shortcut probe (group-aware split, mean±SD over seeds), with a dashed chance line at 0.10. Background-only accuracy clears chance for every family and is largest for the ImageNet-pretrained transfer probe, but every masked condition falls far below the full-image bar, so the dependence on the laboratory canvas is modest and recognition is mostly leaf-driven.

Finally, Table 4 reports a sensitivity sweep of the near-duplicate threshold τ that defines the groups, listing for each value the number of groups recovered and the resulting leakage gap per family. The gap is not an artefact of one threshold: across $\tau = 0.03, 0.06, 0.10$, and 0.15, with the group count moving from 2987 down to 2886, the leakage gap stays inside a band of about ± 0.011 for every family and is frequently negative, for instance -0.0108 for the CNN at the tightest threshold and -0.0053 for the SVM at $\tau = 0.10$. No setting of the threshold uncovers a meaningful gap, which is exactly what we would expect when the corpus contains few same-leaf duplicates to remove.

Table 4: Ablation: sensitivity of the leakage gap to the near-duplicate threshold τ. For each τ we list the number of recovered groups and the leakage gap (naive minus group-aware accuracy) per family. Across the sweep the gap stays inside a ± 0.011 band and is frequently negative; no threshold uncovers a meaningful gap.

τ	Groups	CNN	Transfer	SVM	RF
0.03	2987	−0.0108	+0.0092	0.0000	−0.0004
0.06	2980	+0.0062	+0.0039	+0.0010	−0.0001
0.10	2964	+0.0110	−0.0046	−0.0053	+0.0044
0.15	2886	+0.0040	+0.0040	+0.0001	+0.0122

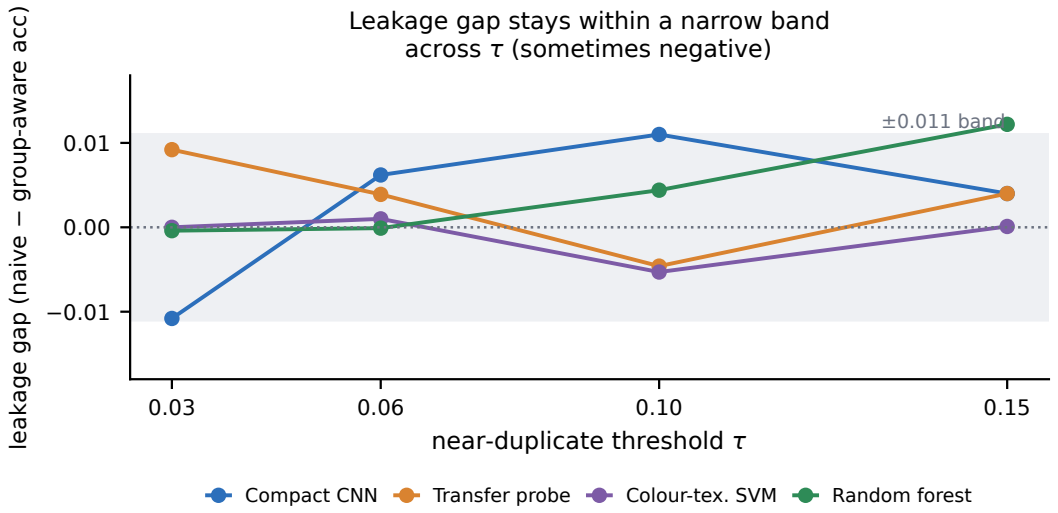

Figure 6: Measured leakage gap (naive minus group-aware accuracy) as a function of the near-duplicate threshold τ, one line per family. The shaded region marks a ± 0.011 band; every family's gap stays inside it and crosses zero, so no threshold uncovers a meaningful near-duplicate leakage gap on this subset.

5 Analysis

5.1 Near-Duplicate Leakage Is Negligible on This Corpus

The central object of this study is a gap, and the measured gap is, against the proposal's expectation, essentially zero. Under the naive split a model could in principle succeed by matching a test image to a near-duplicate of the same source leaf in training, an ability the group-aware split forbids; the drop between the two splits estimates how much apparent accuracy rested on that near-duplicate leakage. Table 2 and Figure 4 show the drop is a fraction of one accuracy point for every family, $+0.0065$ for the compact CNN, $+0.0039$ for the transfer probe, $+0.0010$ for the SVM, and -0.0001 for the random forest, each well inside its seed spread, and Table 4 confirms the gap stays inside a ± 0.011 band and turns negative at several thresholds. The widely assumed large near-duplicate leakage did not materialise here, and the reason is concrete rather than a protocol failure: the colour Tomato subset simply carries very few same-leaf duplicates to leak. Inspecting the full Tomato pool, only about two hundred and ninety-three filename source-leaf provenance keys index more than one view, covering roughly six hundred and fifty images, about 3.6% of the pool, and the capped colour subset inherits even fewer. With almost no near-duplicates straddling the boundary, removing them cannot move the number, so the honest accuracy is essentially the naive accuracy. We read this as a genuine empirical finding: the group-aware protocol behaved correctly, found the few duplicates that exist, and reported truthfully that they do not inflate the result. The reading is consistent with the broader lesson that train/test contamination produces overoptimistic results across machine-learning science [Kapoor and Narayanan, 2023], while sharpening it: a leakage-aware split removes whatever near-duplicate inflation is present, and here that quantity is simply near zero.

This is emphatically not a claim that near-duplicate leakage is harmless in general. It is a statement about one corpus. The leakage gap is dataset-dependent: a benchmark assembled from many augmented copies of each physical specimen, or one whose acquisition revisits the same leaves, would show a wide gap under exactly this protocol, especially for a high-capacity network able to memorise such near-duplicates [Zhang et al., 2021], and the folklore that PlantVillage-style benchmarks leak heavily is plausible for other subsets and other augmentation regimes. What our measurement establishes is that the colour Tomato subset, evaluated honestly, is not inflated by same-leaf recall, and that one cannot assume otherwise without running the audit. The instrument earns its keep precisely by returning a null result here as readily as it would return a large one elsewhere.

5.2 The Background Shortcut Is Real but Modest

The probe in Table 3 and Figure 5 answers a different question: not whether the test leaves were seen, but whether the model reads the lesion at all. Background-only accuracy clears the chance baseline of 0.10 for every family, which confirms a genuine background shortcut of the kind shortcut-learning warns about [Geirhos et al., 2020], amplified by the texture bias of convolutional networks [Geirhos et al., 2019], and the ordering is informative. The transfer probe leans on the canvas the most, reaching 0.231 on background alone, roughly 2.3 times chance, consistent with an ImageNet-pretrained representation that has learned to read global scene statistics; the compact CNN follows at 0.162, the SVM at 0.137, and the random forest at 0.108 is barely above chance. Yet the shortcut is modest, not dominant. Every masked condition collapses toward chance, leaf-only between 0.101 and 0.212 and background-only between 0.108 and 0.231, while the full image yields roughly 0.85 to 0.905, so the overwhelming majority of each family's accuracy comes from the leaf and its lesions rather than from the laboratory background. The pretrained probe is the family to watch, because it both carries an external representation and shows the largest background dependence, but even for it the canvas accounts for a small slice of the full number.

5.3 Deep Versus Classical under an Honest Boundary

Running all four families through the identical honest split lets us ask which is actually strongest once leakage is controlled, and the advocated compact CNN is not the winner. Under the group-aware split the colour-texture SVM led at 0.905 accuracy, the random forest reached 0.888, and the compact CNN at 0.852 sat essentially level with the transfer probe at 0.845. The classical colour-and-texture SVM is the best family on this subset, and we report this plainly rather than implying the deep model won. We report macro-F1 alongside accuracy throughout precisely so that a family cannot win on a dominant class while quietly failing a minority disease a screen exists to catch [Buda et al., 2018]; here accuracy and macro-F1 track each other closely, so the ranking is not an artefact of class imbalance. The comparison carries a caveat we stated in the design: the four families use different input representations and fitting procedures, and the transfer probe in particular carries an external ImageNet representation, so these are each family's own honest number rather than a fully controlled head-to-head. Even read that way, the conclusion is unembellished: a deliberately compact CNN trained from scratch does not exceed a classical colour-texture baseline on this corpus under honest evaluation, and a practitioner choosing a small deployable model has no accuracy reason here to prefer the network over the hand-crafted-feature classifier, and any further compression of the network, for instance toward binary weights [Qin et al., 2020], should be weighed against this honest number rather than a leaky one.

5.4 Failure Modes and Limitations

Several limitations bound how far these readings should be carried, and naming them is part of using the instruments honestly. The first concerns the leakage estimate itself. We biased the near-duplicate detector toward under-merging, so any near-duplicate it misses leaves a leaked image in the test set and lifts the group-aware accuracy; the measured gap is therefore a conservative lower bound on the true leakage rather than a point estimate, and a near-zero measured gap certifies only that the leakage we could detect was small, not that none exists. On this corpus that caveat changes little, because the provenance count already shows few duplicates to miss, but on a denser corpus it would matter. The second concerns the probe. PlantVillage provides no ground-truth leaf masks, so we cannot compute an intersection-over-union against a reference segmentation and do not report one; the segmenter's mean leaf-coverage fraction was 0.370 ± 0.109 with a median of 0.374, an honest coverage proxy accompanied by saved qualitative panels rather than a validated mask quality. Because the mask is imperfect, residual leaf pixels can survive into the background-only image and inflate its accuracy, so the background-only number is best read as an upper-ish estimate of the shortcut, which makes our modest reading conservative in the direction that matters; we also treat synthetic remedies such as generative background augmentation with caution, since they can mask a shortcut rather than remove it [Goodfellow et al., 2020]. The third is the head-to-head caveat already noted: the four families differ in representation, so their absolute accuracies rank families on this subset but do not isolate architecture from input. Finally, the negative leakage result is specific to the colour Tomato subset and three seeds; it should not be generalised into a claim that leakage-aware evaluation is unnecessary, since the whole point of the audit is that one must run it to know.

6 Conclusion

We set out to measure, rather than assume, how much of the near-saturated accuracy reported for leaf-disease classification on PlantVillage is genuine, and we built Leakage-Aware Leaf Evaluation to do it: a group-aware split that forbids near-duplicates of a training leaf at test time, a sweep over the near-duplicate threshold, and a three-condition probe that masks the leaf or the background. The contribution is this evaluation methodology together with the honest, partly negative findings it returns on the colour Tomato subset. The widely repeated worry about near-duplicate leakage did not hold here: the leakage gap was a fraction of an accuracy point for every family and stayed within a narrow band across the threshold sweep, because the subset carries few same-leaf duplicates to leak, a result that is dataset-dependent rather than a verdict that leakage never matters. The background shortcut was real but modest, largest for the ImageNet-pretrained transfer probe, with recognition remaining overwhelmingly leaf-driven, and the compact CNN did not beat the classical colour-texture baselines under the honest split. The study's reach is bounded by the conservative under-merging of the near-duplicate detector and by the absence of ground-truth masks, which is why we report a coverage proxy rather than an intersection-over-union and treat the shortcut estimate as an upper-ish one. The natural next step is to carry the same audit to denser and to in-the-wild corpora, where the very confounds that proved negligible here may instead dominate, and where only running the audit can tell.

References

Amreen Abbas, Sweta Jain, Mahesh Gour, and Swetha Vankudothu. Tomato plant disease detection using transfer learning with c-gan synthetic images. *Computers and Electronics in Agriculture*, 187:106279, 2021. doi: 10.1016/j.compag.2021.106279.

Marko Arsenovic, Mirjana Karanovic, Srdjan Sladojevic, Andras Anderla, and Darko Stefanovic. Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry*, 11(7):939, 2019. doi: 10.3390/sym11070939.

Jayme Garcia Arnal Barbedo. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153:46–53, 2018. doi: 10.1016/j.compag.2018.08.013.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. doi: 10.1016/j.neunet.2018.07.011.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011. doi: 10.1145/1961189.1961199.

Junde Chen, Jinxiu Chen, Defu Zhang, Yuandong Sun, and Y. A. Nanehkaran. Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173:105393, 2020a. doi: 10.1016/j.compag.2020.105393.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint*, 2020b.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Mads Dyrmann, Henrik Karstoft, and Henrik Skov Midtiby. Plant species classification using deep convolutional neural network. *Biosystems Engineering*, 151:72–80, 2016. doi: 10.1016/j.biosystemseng.2016.08.024.

Konstantinos P. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018. doi: 10.1016/j.compag.2018.01.009.

G. Geetharamani and Arun Pandian J. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers and Electrical Engineering*, 76:323–338, 2019. doi: 10.1016/j.compeleceng.2019.04.011.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint*, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. doi: 10.1145/3422622.

Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint*, 2017.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *arXiv preprint*, 2017.

David P. Hughes and Marcel Salathé. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint*, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint*, 2015.

Andreas Kamilaris, Andreas Kartakoullis, and Francesc X. Prenafeta-Boldú. A review on the practice of big data analysis in agriculture. *Computers and Electronics in Agriculture*, 143:23–37, 2017. doi: 10.1016/j.compag.2017.09.037.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. doi: 10.1145/3065386.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539.

Lili Li, Shujuan Zhang, and Bin Wang. Plant disease detection and classification by deep learning—a review. *IEEE Access*, 9:56683–56698, 2021. doi: 10.1109/ACCESS.2021.3069646.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6):1–35, 2021. doi: 10.1145/3457607.

Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016. doi: 10.3389/fpls.2016.01419.

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. doi: 10.1109/TPAMI.2002.1017623.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint*, 2017.

Artzai Picon, Aitor Alvarez-Gila, Maximiliam Seitz, Amaia Ortiz-Barredo, Jone Echazarra, and Alexander Johannes. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Computers and Electronics in Agriculture*, 161:280–290, 2019. doi: 10.1016/j.compag.2018.04.002.

Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *Pattern Recognition*, 105:107281, 2020. doi: 10.1016/j.patcog.2020.107281.

Amanda Ramcharan, Kelsee Baranowski, Peter McCloskey, Babuali Ahmed, James Legg, and David P. Hughes. Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8:1852, 2017. doi: 10.3389/fpls.2017.01852.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. Plant disease detection and classification by deep learning. *Plants*, 8(11):468, 2019. doi: 10.3390/plants8110468.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. doi: 10.1186/s40537-019-0197-0.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, 2015.

Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. PlantDoc: A dataset for visual plant disease detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 249–253. ACM, 2020. doi: 10.1145/3371158.3371196.

Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016:3289801, 2016. doi: 10.1155/2016/3289801.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594.

Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. *arXiv preprint*, 2019.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *arXiv preprint*, 2016.

Edna Chebet Too, Yujian Li, Sam Njuki, and Yingchun Liu. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161: 272–279, 2019. doi: 10.1016/j.compag.2018.03.032.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy: FixEfficientNet. *arXiv preprint*, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. doi: 10.1145/3446776.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021. doi: 10.1109/JPROC.2020.3004555.