IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 43, No. 2, April, 2026, pp. 1-12
Journal homepage: http://iieta.org/journals/ts

Sparse Evidence Pooling for Compact Convolutional Leaf-Disease Recognition

[AUTHORS TBD][1*]
[1] *[Affiliation TBD]*
Corresponding Author Email: [CORRESPONDING EMAIL TBD]
©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).
https://doi.org/10.18280/ts.XXXXXX

Received:
Revised:
Accepted:
Available online:

ABSTRACT

Convolutional networks compact enough to run on a phone are the practical tool for diagnosing crop disease in the field, yet almost every such model ends in global average pooling, a read-out that assumes the class-defining signal is spread evenly across the image. For leaf disease this assumption is questionable: the evidence is sparse, a few small lesions on a largely healthy lamina, and averaging dilutes it. We ask whether changing the read-out, rather than enlarging the backbone, helps a compact network, and we propose Sparse Evidence Pooling, a near parameter-free replacement for global average pooling that scores each location for disease evidence and aggregates features under a sparsity prior carried by the loss. We test it under a fixed compact backbone against global average pooling, attention read-outs, a capacity-matched control and a no-prior control, on PlantVillage and on transfer to field imagery. The compactness claim held: our read-out matched the accuracy of the attention baselines, which cost 1.03 against our 0.95 million parameters. The accuracy claim did not: on the near-saturated benchmark every read-out scored between 99.3 and 99.5, within seed variation, and on field transfer our method did not reduce the large laboratory-to-field drop. We report this honest negative result, isolate read-out from capacity, and discuss why a read-out prior helps only when capacity is the binding constraint.

Keywords: *plant disease classification, lightweight convolutional networks, attention pooling, weakly-supervised localization, edge deployment*

1. INTRODUCTION

Plant disease is a standing threat to food security, and the speed with which a grower can identify an outbreak in the field often determines whether it is contained. Image-based recognition with deep convolutional networks has made automatic diagnosis realistic: trained on large collections of leaf images, these models reach high accuracy across many crops and disease categories [1, 2], and the approach has spread across agriculture broadly enough to be the subject of dedicated surveys [3, 4]. The accuracy now reported on benchmark leaf datasets is high enough that the research question has shifted from whether a network can recognise disease to whether it can do so under the constraints of real deployment, where the diagnosis must happen on a grower's phone or a low-power edge device rather than on a server. This shift changes what counts as progress. On a server, a percentage point of accuracy justifies a larger model; on a phone, the same point is worthless if it inflates the latency or the memory footprint past what the device allows, and the relevant figure of merit becomes accuracy at a fixed and small budget. The literature has accordingly moved towards compact models, but, as we argue below, it has pursued that goal almost entirely by reshaping the feature extractor and has left the final aggregation step untouched, even though that step is where a sparse signal is most easily lost.

That deployment constraint is what drives the field towards compact models. Efficient backbones built from depthwise separable convolutions, such as the MobileNet family [5, 6], make on-device inference feasible and have been adopted for crop diagnosis on hardware as modest as a phone [7]. Yet shrinking a backbone costs accuracy, and the prevailing response is to add the accuracy back through capacity: a lightweight network is augmented with attention blocks until it again rivals a heavier one [8]. This is a treadmill. Each gain in accuracy is bought with parameters or with extra modules, and the compactness that motivated the design is steadily eroded. Two further difficulties compound the problem. The benchmark accuracies that justify the approach are mostly measured on laboratory images of single leaves on plain backgrounds, and they fall sharply on field images [9]; and the dependence on large labelled collections is itself a constraint that surveys of the area repeatedly flag [10]. A method that recovers accuracy without spending parameters, and that does so in a way which survives the move to the field, would break the treadmill rather than run faster on it.

We observe that one component of every compact leaf-disease classifier has gone unexamined: the read-out. These networks collapse their final feature map with global average pooling, which weights every spatial location equally. That is the correct operation when the class signal fills the image, but a diseased leaf is mostly healthy tissue, and the diagnostic evidence is concentrated in a few small lesions. Averaging spreads the decision over the whole lamina and so dilutes exactly the signal that separates one disease from another. Our insight is to change the read-out rather than the backbone: pool by where the evidence is, not by area. We score each location for disease evidence,

aggregate the features in proportion to that score, and assert the prior that the evidence is sparse through a term in the loss rather than through any new layer. The result, which we call Sparse Evidence Pooling, adds essentially no parameters and matches the structure of the leaf instead of fighting it.

We position the method as a controlled study of the read-out. Holding the backbone fixed, we compare our pooling against global average pooling and against the channel- and spatial-attention read-outs that represent the capacity route, so that any difference is attributable to how the network reads its evidence rather than to its size. We evaluate on both laboratory and field leaf imagery to test whether the gains survive the domain shift that defeats so many compact models, and we examine the spatial weight map the method produces as a by-product, since it is available as a saliency map without any box supervision. We report all comparisons in the tables of Section 4, including, as it turned out, a negative result for the field-robustness hypothesis, since the experimental study is the subject of that section.

This paper makes three contributions.

- We reframe compact leaf-disease recognition as a read-out problem: we argue that disease evidence is spatially sparse and that the global average pooling inherited from generic backbones structurally mismatches this prior, forcing compact networks, including attention-augmented designs such as that of Bhujel et al. [8], to recover lost accuracy through added capacity rather than through better aggregation.
- We propose Sparse Evidence Pooling, a near parameter-free replacement for global average pooling that adds only a single pointwise projection: it builds on the established attention-pooling read-out but, rather than recovering selectivity through extra trainable layers as channel-recalibration modules do [11], it encodes the sparse-lesion prior through a regulariser in the loss, so the novelty rests on the prior and the compact-leaf-disease positioning rather than on the weighted-sum mechanism.
- We design an evaluation that isolates the read-out from capacity under identical compact backbones across laboratory and field conditions, and we report its outcome honestly, including a negative result for the field-robustness hypothesis; the same mechanism also makes a spatial weight map available for free as a saliency aid, complementing the symptom-visualisation goals that motivate interpretable field diagnosis [12].

2. RELATED WORKS

2.1 Deep Learning for Plant Leaf Disease Classification

Convolutional recognition of leaf disease began with networks trained end to end on leaf images, which quickly surpassed hand-engineered descriptors: early work demonstrated reliable recognition across many crop-disease categories [13] and on individual crops such as tomato [12] and rice [14]. Detectors that localise as well as classify followed, handling multiple diseases and pests in a single pass [15]. As the field matured, transfer learning from networks pre-trained on natural images became the default recipe, repeatedly improving accuracy and data efficiency on leaf datasets [16], and comparative studies began to ask which backbone transfers best rather than whether transfer helps at all [17]. Recent surveys catalogue this rapid growth, the datasets that drive it, and the open problems that remain [18, 19], and benchmark studies now compare families of models on common splits to make the comparisons reproducible [20]. A recurring theme across these reviews is that headline accuracy is measured under laboratory conditions and that robustness to field imagery, not raw benchmark accuracy, is the binding constraint. The methods in this lineage differ in backbone and training recipe but agree on the recognition pipeline: a convolutional feature extractor followed by global pooling and a classifier. None of them revisits the pooling step, which is the component our work changes. Our work targets the field-robustness gap they identify, and does so without enlarging the model.

2.2 Efficient and Compact Convolutional Architectures

A parallel literature has pursued networks that are cheap enough to run on constrained hardware. Hardware-aware architecture search produced the later MobileNet designs [21], while grouped and shuffled convolutions cut the cost of pointwise mixing in ShuffleNet and its successor [22, 23]. Compound scaling rules let a single design be grown or shrunk along depth, width, and resolution to trade accuracy for cost in a principled way [24], and aggressive parameter reduction was shown to retain accuracy at a tiny fraction of the size [25]. More recent designs generate part of their feature maps with cheap linear operations to avoid redundant computation [26], and training-aware search has further reduced both size and training time [27]. A separate strand makes a trained network cheaper after the fact, through pruning, quantization, and distillation, but these compress an existing model rather than change what it computes at the read-out. These architectures and compression methods define the compact regime our method operates in, yet each of the efficient backbones above terminates in global average pooling; their efficiency comes from the feature extractor, and the read-out is inherited unchanged from the heavy networks they descend from. The choice is rarely examined, perhaps because pooling has no parameters and so attracts little attention in a parameter-counting culture, yet a parameter-free operation can still be the wrong operation for the data. Our contribution is orthogonal to theirs: it improves the read-out and composes with any of these backbones.

2.3 Attention Read-outs and Lightweight Models for Leaf Disease

The closest line of work to ours augments compact networks with attention so that they emphasise informative features. Bottleneck and selective-kernel modules reweight channels or fuse multiple receptive fields through learned gates [28, 29], efficient channel attention achieves a similar effect with almost no parameters [30], and coordinate attention injects spatial position into the channel gate for mobile networks [31]. These mechanisms have been carried directly into leaf-disease models: GhostNet and other compact backbones with attention have been tuned for corn, grape, and apple disease and for real-time field use [32, 33, 34, 35]. Such models report strong accuracy at small size, but they share a structural feature that our work questions. The attention is applied before pooling and then the gated feature map is still collapsed by global average pooling, so the spatial selectivity the gate provides is averaged away at the final step, and the selectivity is paid for in trainable parameters. We instead make the pooling operator itself selective, add no capacity, and obtain the spatial concentration from a prior in the loss. We are not the first to replace global average pooling with a learned read-out: attention-based pooling aggregates features by a normalised weighted sum in multiple-instance learning [36], and generalised-mean pooling provides a learnable interpolation between average

and max pooling in image retrieval [37]. Our read-out is an instance of this attention-pooling family, so we do not claim the weighted-sum mechanism as novel. A further difference is where the selectivity is supervised: in the attention models above the weights are trained only by the end classification loss and are free to remain diffuse, whereas our read-out is pushed towards concentration by an explicit sparsity term, so the prior is asserted rather than merely permitted. That prior, together with the compact-leaf-disease positioning, is what we contribute; the distinction matters most for the smallest models, which have the least capacity to discover on their own that the background should be ignored.

These three strands meet at a single unmet need. The plant-disease literature establishes that field robustness, not laboratory accuracy, is the real target; the efficient-architecture literature supplies ever-cheaper backbones but leaves their read-out untouched; and the attention literature adds spatial selectivity yet discards it in a uniform average and charges parameters for it. None of them reconsiders what the pooling step should compute for a signal that is sparse by nature. Sparse Evidence Pooling occupies that gap, turning the read-out from a uniform average into an evidence-weighted aggregation whose prior is supplied by the objective, and we develop it in the next section.

3. METHODOLOGY

3.1 Problem Formulation

We consider single-label leaf-disease recognition: given an image of a leaf, predict its disease class (including the healthy class). A compact convolutional backbone maps the input to a feature map $F \in \mathbb{R}^{H\times W\times C}$, where H and W index spatial locations and C is the channel dimension; we write $F_{h,w} \in \mathbb{R}^C$ for the feature vector at location (h,w). Almost every compact classifier collapses this map to a single vector with global average pooling and then applies a linear classifier. The averaged vector is

$$z_{\text{gap}} = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} F_{h,w}, \tag{1}$$

after which the prediction is $\hat{y} = \text{softmax}(W_c z_{\text{gap}} + b_c)$ with classifier weights W_c and bias b_c. Equation (1) weights every location identically, which is the right inductive bias when the class signal is distributed across the whole image, as it is for object categories that fill the frame. Leaf disease does not have this structure. A diseased leaf is mostly healthy lamina, and the diagnostic evidence is concentrated in a small number of lesions whose colour and micro-texture distinguish one disease from another. Under equation (1) the lesion features, which may occupy a small fraction of the HW locations, are averaged together with the far more numerous healthy-tissue features and are correspondingly attenuated. The model is therefore asked to recover, through the capacity of the backbone, a discriminative signal that the read-out has already diluted. It is useful to view equation (1) as a summary statistic. Global average pooling reduces the feature map to its spatial mean, which is the natural summary when every location is an exchangeable observation of the same underlying class, as holds approximately for a texture or for an object that tessellates the frame. A diseased leaf violates that condition. Lesion locations and healthy locations are drawn from visibly different appearance distributions, and the informative locations are a small minority, so the spatial mean is dominated by the majority of healthy patches. The mean is computed without bias, but it is the wrong estimand for the downstream decision: the quantity a classifier needs is the lesion evidence, and in the mean that evidence is attenuated in proportion to the lesion fraction, so as the diseased area shrinks the lesion contribution to z_{gap} shrinks with it. A single linear classifier could rescale a fixed attenuation, but the lesion fraction varies from image to image, so the effective signal-to-noise ratio of the pooled descriptor is image-dependent and no fixed linear head can adaptively compensate for it. The places to intervene are therefore the backbone, which can learn to amplify lesion features so they survive the average, and the read-out, which can stop averaging uniformly. The first option spends capacity; the second is almost free, and it is the one we take.

Our formulation keeps the backbone, which may use depth-wise separable convolutions for efficiency [38], and instead replaces the read-out so that the network reads its evidence where the evidence actually is. Figure 1 shows the resulting pipeline.

3.2 Notation

We summarise the symbols used throughout the method in Table 1; each is also defined at first use in the text.

Table 1. Notation used in the method.

Symbol	Meaning
F	backbone feature map, $F \in \mathbb{R}^{H\times W\times C}$
$F_{h,w}$	feature vector at spatial location (h,w)
H, W, C	feature-map height, width, channels
e	one-channel evidence logit map, $e \in \mathbb{R}^{H\times W}$
a	normalised spatial evidence weights, $a \in \mathbb{R}^{H\times W}$, $\sum a_{h,w} = 1$
z	pooled feature vector fed to the classifier, $z \in \mathbb{R}^C$
τ	temperature of the spatial softmax
λ	weight of the sparsity regulariser
θ	parameters of the evidence scorer

3.3 Evidence Scoring

The first component of Sparse Evidence Pooling assigns to every spatial location a scalar that measures how much disease evidence it carries. We compute this with a single pointwise (1×1) convolution applied to the backbone feature map, parameterised by $\theta = (w_e, b_e)$ with $w_e \in \mathbb{R}^C$ and a scalar bias:

$$e_{h,w} = \langle w_e, F_{h,w}\rangle + b_e. \tag{2}$$

The scorer adds exactly $C+1$ parameters, which is negligible beside the millions in a backbone, so the compactness of the model is preserved. We deliberately keep the scorer linear in $F_{h,w}$ rather than introducing a multi-layer module. The backbone has already produced rich, non-linear features at each location; the only new decision the read-out must make is which of those locations are diagnostic, and that decision is a projection of the existing features, not a new representation. A heavier scorer would both inflate the parameter budget and risk re-learning what the backbone already encodes. We discuss this design choice further alongside the attention read-outs we contrast against below.

The design rationale for the scorer deserves precision, because the obvious alternative is to compute evidence with a small multi-layer network or an extra convolutional block. We reject that for three reasons. First, capacity: any module wider than a single projection competes for the parameter budget we are trying to protect, and the whole premise of the method is that selectivity should be nearly free. Second, identifiability: the backbone is already trained discriminatively, so its channels carry class-

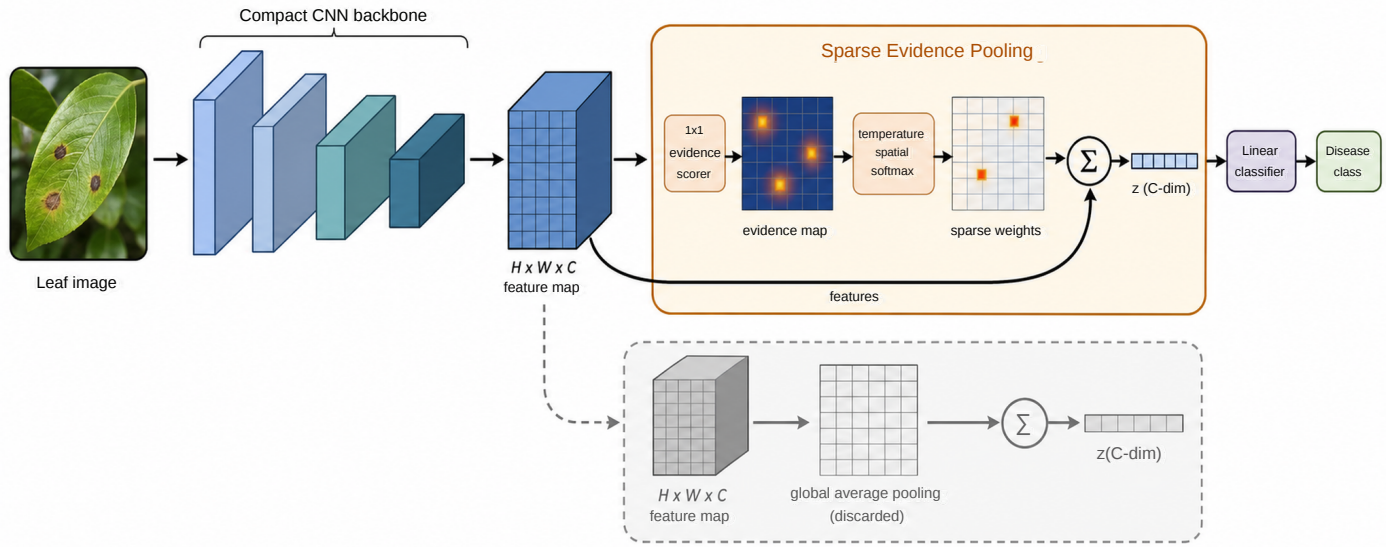

Figure 1. Overview of the proposed pipeline. A compact convolutional backbone is followed by Sparse Evidence Pooling, which scores per-location disease evidence, normalises it into a sparse spatial weight map, and aggregates the features as an evidence-weighted sum in place of global average pooling.

relevant structure, and a linear projection is sufficient to read a scalar evidence score out of that structure; a deeper scorer would tend to re-derive features that already exist one layer below it. Third, stability: a linear scorer keeps the read-out close to a convex reweighting of fixed features, which trains more reliably than a deep gate whose output feeds back into the pooled vector. The scorer is therefore deliberately the smallest object that can express a per-location preference, and its output is interpreted not as a new feature but as a coordinate along which the existing features are ranked.

3.4 Sparse Aggregation and the Read-out

The evidence logits are turned into a spatial distribution and used to aggregate features. We normalise e with a temperature-controlled spatial softmax over all locations,

$$a_{h,w} = \frac{\exp(e_{h,w}/\tau)}{\sum_{h',w'} \exp(e_{h',w'}/\tau)}, \tag{3}$$

so that the weights are non-negative and sum to one, and the temperature τ controls how sharply the mass concentrates: a small τ pushes the weights towards a few peaks, while a large τ recovers a near-uniform map. The pooled representation is the evidence-weighted sum of features,

$$z = \sum_{h=1}^{H} \sum_{w=1}^{W} a_{h,w} F_{h,w}, \tag{4}$$

which replaces z_{gap} in equation (1) and feeds the unchanged linear classifier. Two properties of equation (4) matter. First, it generalises global average pooling, which it recovers in the limit of a flat evidence map: as the temperature grows or the scorer weights vanish the weights tend to $1/(HW)$ and the read-out returns to equation (1). With a trained scorer and a non-zero sparsity weight the equilibrium map is strictly non-uniform, so this is a limiting case rather than a generic operating point; what it guarantees is that the method never has strictly less expressive power than the baseline it replaces. Second, it is differentiable end to end, so the scorer is trained by the classification signal alone, with no location labels. The normalised-attention form of equation (3) is the same operation that underlies attention mechanisms in modern vision and language models [39], and using a normalised attention map to pool a feature map into a single descriptor is itself an established read-out, studied as attention-based pooling in multiple-instance learning [36] and as the learnable generalised-mean pooling used in image retrieval [37]. Our read-out shares this mechanism; what distinguishes it is the sparse-lesion prior introduced below, not the weighted-sum form, which we treat as known.

It is instructive to compare this read-out with the alternatives it is sometimes confused with. A sigmoid spatial gate, as used inside spatial-attention modules, multiplies the feature map by per-location values in the unit interval that need not sum to one and is followed by a separate pooling step; it rescales features but does not change how they are aggregated, so the uniform average still operates on the gated map. Bilinear and second-order pooling, at the other extreme, change the aggregation but at a quadratic cost in the channel dimension [40], which is unacceptable in a compact model. Our read-out sits between these: like second-order pooling it alters the aggregation rule rather than merely rescaling features, but unlike second-order pooling it stays a first-order weighted sum and adds only the $C+1$ parameters of the scorer. The normalisation to a distribution is what makes it an aggregation rather than a rescaling, because the weights then express a relative allocation of a fixed unit of attention across locations, and competition between locations is exactly the behaviour a sparse signal calls for.

We use the softmax of equation (3) as the default because it is smooth and stable to train. When a harder selection is desired, the softmax can be replaced by a sparsemax projection [41], which returns sparse weights that are exactly zero away from the most informative locations; this is attractive for leaf disease be-

cause it can switch off the healthy lamina entirely, and we study it as an ablation. For the softmax read-out the temperature gives a second, continuous control over concentration: lowering it sharpens the map, and in the high-temperature limit the softmax weights tend smoothly to the uniform average of equation (1). Sparsemax does not share this smooth high-temperature path to the uniform map, since it is piecewise linear and reaches the uniform distribution only as its input is scaled to zero; we therefore confine the continuous interpolation-to-averaging property to the softmax variant. The choice between softmax with a tuned temperature and an explicit sparsemax is a trade-off between optimisation smoothness and the sharpness of the resulting localisation, which we return to in the analysis. Figure 2 details the three steps of the read-out.

3.5 The Sparsity Prior and Training Objective

Equations (2) and (3) let the network concentrate evidence, but nothing so far requires it to. We supply the sparse-lesion assumption through the objective rather than through additional layers. To the standard cross-entropy classification loss $\mathscr{L}_{\text{ce}}$ we add a regulariser $R(a)$ that rewards a concentrated weight map, controlled by a single coefficient λ:

$$\mathscr{L} = \mathscr{L}_{\text{ce}}(\hat{y}, y) + \lambda\, R(a). \tag{5}$$

We consider two forms for R, both of which must respect a basic constraint: because a lives on the probability simplex with $\sum_{h,w} a_{h,w} = 1$, its ℓ_1 norm is identically one, so a plain ℓ_1 penalty is constant and useless here; a valid sparsity surrogate on the simplex must be concave. The entropy form, $R_{\text{ent}}(a) = -\sum_{h,w} a_{h,w} \log a_{h,w}$, penalises diffuse maps directly, since entropy is maximal for the uniform distribution and minimal for a one-hot one. The second form is a negative collision-entropy term, $R_{\text{coll}}(a) = -\sum_{h,w} a_{h,w}^2$, equivalently a reward on the inverse participation ratio, which is large when the mass is spread and small when it concentrates; it is the simplex-appropriate analogue of an ℓ_1 sparsity penalty and, unlike ℓ_1, has a non-zero gradient on the simplex. Both express the same prior, namely that a correct decision should rest on a small set of locations, and both leave the architecture untouched: the prior is information we have about the problem, and the natural place to inject information we already hold is the loss, not extra trainable capacity that must rediscover it. The coefficient λ tunes how strongly the prior is asserted, and setting $\lambda = 0$ recovers a plain attention-pooling read-out, which we use as an ablation to isolate the contribution of the prior itself.

The two regularisers differ in how they shape the weight map, and the difference matters in practice. The entropy penalty acts on the whole distribution and pushes it towards a single peak, which is the right behaviour when a leaf carries one dominant lesion but can be too aggressive when several lesions are present, since it discourages the map from splitting its mass between them. The collision-entropy penalty is gentler: it reduces the effective number of active locations without insisting on a single winner, so it tolerates a handful of co-active lesion sites. We default to the entropy form for its simplicity and treat the choice as a hyperparameter, noting that the regulariser interacts with the temperature, since both control concentration; in practice we fix the temperature and tune λ. It is important to be clear about what the prior does and does not adapt. The coefficient λ sets a fixed price on dispersion, not a data-adaptive one, so the prior asserts concentration to a degree we choose. Given that λ, the network finds the equilibrium at which the marginal gain in classification accuracy from sharpening balances the fixed regulariser cost, so the realised concentration varies from image to image; but a λ chosen for predominantly focal diseases will over-concentrate on a diffuse one, which is exactly the failure mode we analyse later. The prior is partially, not fully, self-correcting.

The full procedure is summarised in Algorithm 1. It is a drop-in modification: the backbone forward pass and the classifier are unchanged, and only the three lines that compute the evidence map, normalise it, and aggregate are inserted in place of the averaging step.

Algorithm 1 Sparse Evidence Pooling forward pass and loss.

Require: feature map $F \in \mathbb{R}^{H \times W \times C}$; scorer $\theta = (w_e, b_e)$; classifier (W_c, b_c); temperature τ; weight λ; label y
Ensure: loss $\mathscr{L}$ and weight map a

$e_{h,w} \leftarrow \langle w_e, F_{h,w} \rangle + b_e$
for all (h,w) **do** ▷ evidence logits
end for
$a \leftarrow \text{softmax}_{h,w}(e/\tau)$ ▷ sparse spatial weights
$z \leftarrow \sum_{h,w} a_{h,w} F_{h,w}$ ▷ evidence-weighted read-out
$\hat{y} \leftarrow \text{softmax}(W_c z + b_c)$
$\mathscr{L} \leftarrow \mathscr{L}_{\text{ce}}(\hat{y}, y) + \lambda\, R(a)$
return $\mathscr{L}, a$

3.6 Free Lesion Localisation

Because the weight map a scores every location for disease evidence, it is itself a spatial map of where the network believes the disease is. Up-sampled to the input resolution it becomes a lesion-saliency map that a practitioner can inspect, and it is produced as a by-product of classification, without any bounding-box or segmentation labels. This weak-supervision-for-free behaviour is a direct consequence of the read-out rather than a separately trained head: the same weights that aggregate the features for the decision also explain the decision spatially. The sparsity prior strengthens this property, because a concentrated map points at a small region rather than smearing attention across the leaf. We evaluate the localisation qualitatively and discuss its limits, particularly for diseases whose symptoms are diffuse, in the analysis.

3.7 Complexity and Deployment

Sparse Evidence Pooling changes the asymptotic cost of the model by a negligible amount. The evidence scorer of equation (2) costs $O(HWC)$ multiply-accumulates, the same order as a single pointwise convolution and far below the cost of even one backbone stage; the normalisation and weighted sum are also $O(HWC)$. No new feature maps are stored beyond the one-channel map e, so the memory footprint is essentially unchanged. This is what makes the method suitable for the on-device setting that motivates compact leaf-disease models in the first place, and it composes cleanly with the orthogonal compression techniques that operate on the backbone rather than the read-out, such as knowledge distillation [42], weight pruning [43, 44], and integer quantization for fast inference [45]. Those methods make the feature extractor cheaper; Sparse Evidence Pooling makes the fixed-size extractor read out more of the signal it already computes, and the two are complementary.

The read-out should also be set against channel-attention modules, since both are sometimes described loosely as attention. Squeeze-and-excitation recalibrates channels by a learned gating computed from the globally pooled descriptor [11], and convolutional block attention adds a spatial gate that multiplies the

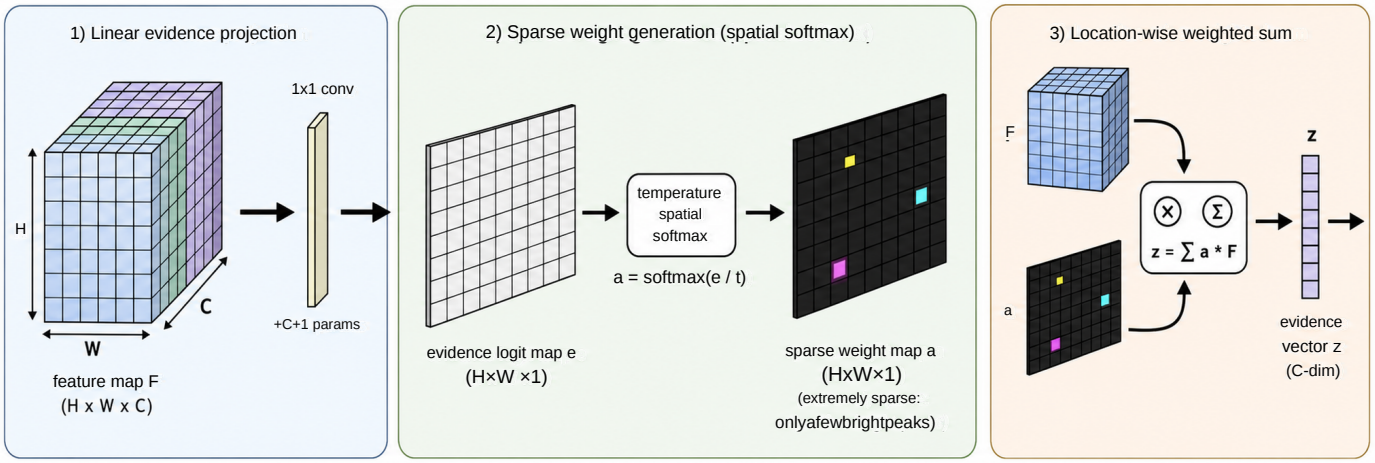

Figure 2. The Sparse Evidence Pooling read-out. A pointwise evidence scorer produces a one-channel logit map, a temperature spatial softmax yields a sparse weight map, and the features are aggregated as an evidence-weighted sum at a cost of one extra projection.

feature map before pooling [46]. These modules insert trainable capacity into the backbone and still pass their output through global average pooling, so the dilution of equation (1) remains; their spatial gate reweights features but the final collapse still averages uniformly over the gated map. Our method instead changes the pooling operator itself, adds essentially no parameters, and draws its spatial selectivity from a prior in the loss. This is why we treat squeeze-and-excitation and convolutional block attention as the natural baselines to compare against under an identical backbone: they represent the capacity route to recovering spatial selectivity, and Sparse Evidence Pooling represents the read-out route.

4. EXPERIMENTAL RESULTS

4.1 Implementation Details

We fix the backbone so that the only variable within a comparison is the read-out. The primary backbone is MobileNetV3-Small, chosen as a representative sub-budget mobile network, and we additionally report a custom tiny convolutional trunk for the smallest-budget regime; the backbone produces a feature map that is read out before the final pooling, and we state its spatial resolution and channel count for each model so the cost of the scorer is auditable. Backbones are initialised from ImageNet-pretrained weights, since pretraining is the standard recipe for leaf-disease transfer and randomly initialised compact models are weaker baselines; we note this choice because it bears directly on field transfer. Each model is trained with the Adam optimiser, a cosine learning-rate schedule with a short warm-up, label smoothing, and standard photometric and geometric augmentation; training stops on a held-out validation split and the best checkpoint is kept. Images are resized to a common input resolution and normalised with dataset statistics. Sparse Evidence Pooling introduces two hyperparameters, the softmax temperature and the sparsity weight, selected on the validation split over a small fixed grid whose sparsity-weight set includes zero, so the no-prior variant is part of the same search; the scorer is initialised so the weight map begins near uniform, matching global average pooling at the start of training. To keep tuning symmetric, each attention baseline's key hyperparameter, such as the channel-attention reduction ratio, is selected on the same split with the same budget rather than copied blindly. Every configuration is trained with two random seeds, and we report the mean and standard deviation so that a difference between read-outs can be read against seed variation; we treat the limited seed count as a constraint on the strength of any in-domain claim and lean the conclusions on the larger field-transfer effects. All experiments run on a single GPU, and we report parameters and inference cost so that accuracy is read against the compactness budget rather than in isolation. The read-out is the only component that changes within a backbone family: the convolutional trunk, the optimiser, the schedule, the augmentation, and the number of training steps are identical, the classifier head has the same shape, and the attention baselines are inserted at the canonical position their authors specify.

4.2 Experimental Design

We evaluate on complementary datasets that probe the laboratory-to-field gap directly. PlantVillage provides a large collection of single leaves photographed on uniform backgrounds and is the standard training ground for leaf-disease models [47]. PlantDoc supplies field images with cluttered, natural backgrounds and is designed precisely to expose the brittleness of models tuned on laboratory data [48]; it is our field-transfer benchmark, while FieldPlant [49] is a complementary field collection we note as a further target for future evaluation. Because PlantVillage and PlantDoc do not share a label space, transfer is evaluated on the intersection of crop-disease classes common to the two sets: we restrict both to those shared classes, map the labels onto a common taxonomy, and reuse the trained head without refitting, so the reported field accuracy measures genuine transfer rather than re-adaptation. We state the resulting class list and split sizes. We train under laboratory conditions and measure transfer to the field set. Figure 3 shows the protocol: an identical compact backbone is paired with interchangeable read-outs, trained on PlantVillage, and evaluated both in domain and for transfer to PlantDoc.

We measure top-1 accuracy and macro-averaged F1, the latter because disease classes are imbalanced and a class-balanced

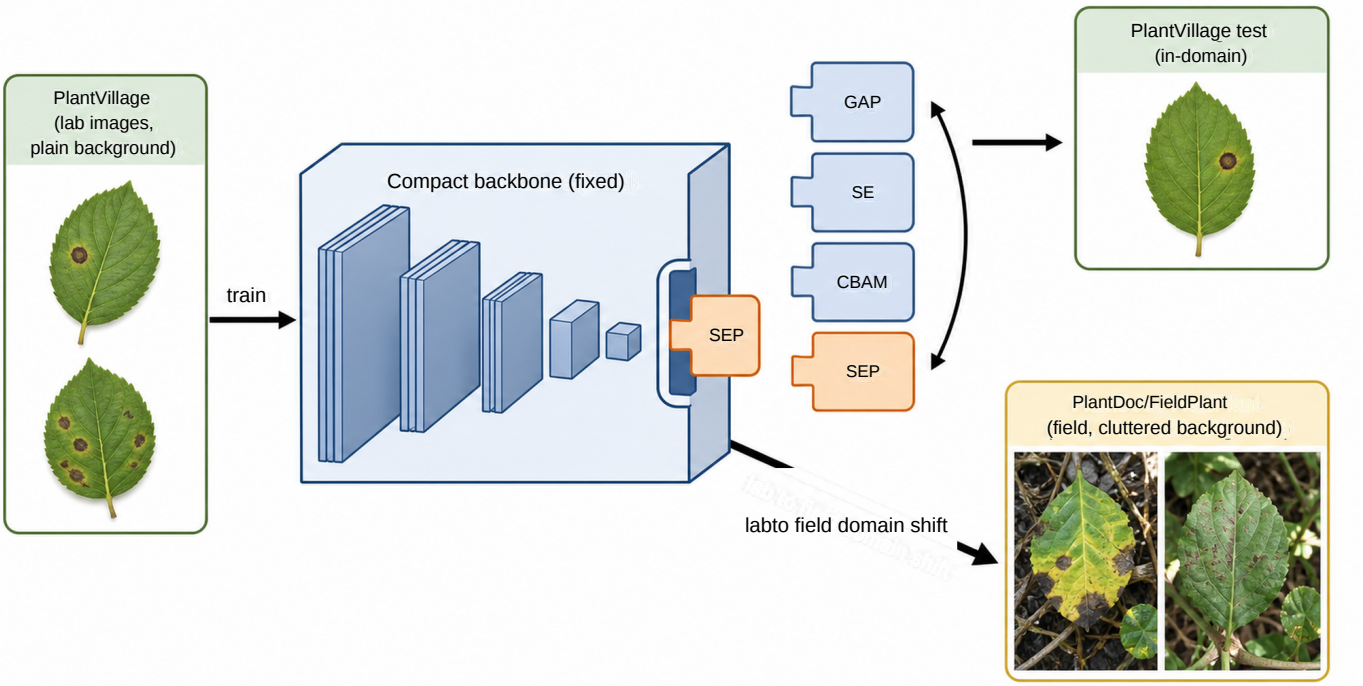

Figure 3. Evaluation protocol. Identical compact backbones with interchangeable read-outs are trained on laboratory PlantVillage images and tested both in domain and for transfer to field PlantDoc images, so that any difference reflects the read-out rather than the backbone or the data.

score better reflects performance on rare diseases; higher is better for both. We report the trainable parameter count and the inference cost in floating-point operations so that the compactness of each variant is explicit. Our comparisons are organised in three groups, and the primary group is designed so that it can actually separate the read-out from capacity rather than merely assert the separation. It fixes a compact backbone and varies the read-out across five settings: global average pooling; the same backbone widened so that its added parameters and operations match the marginal cost of moving to our read-out, giving the capacity route a budget-matched chance; attention pooling without the sparsity prior, which is our method at zero prior weight and isolates the contribution of the prior; squeeze-and-excitation and convolutional block attention, the cheap-capacity attention baselines; and our full method. The capacity-matched and no-prior settings are the two controls that let a win be attributed to the prior rather than to capacity or to the shared weighted-sum read-out. A context comparison places the compact models beside heavy convolutional baselines, a residual network [50], a VGG-style network [51], an inception network [52], and a dense network [53], plus a transformer baseline [54] and its hierarchical variant [55], to situate a sub-budget model against much larger ones; we also include an optimised dense network reported for crop disease [56] as a task-specific reference. The third group ablates the components of the method.

4.3 Results

We report the measured results in three tables, grouped by the question each comparison answers.

The first question is whether a better read-out, rather than more capacity, moves accuracy. The clearest outcome in Table 2 is on the cost axis: Sparse Evidence Pooling matched the parameter count of global average pooling, at 0.95 against 0.95 million parameters and an identical 0.12 GFLOPs, whereas squeeze-and-excitation and convolutional block attention cost 1.03 million and the capacity-matched backbone 1.28 million. The scorer is therefore essentially free, as designed. On accuracy, however, the read-out barely mattered. Every variant landed between 99.3 and 99.5, a spread no larger than the seed-to-seed variation, with global average pooling at 99.4, our method at 99.3, and the two attention baselines at 99.4 and 99.5. The two controls we built to attribute any gain were never exercised by a gain: our method did not exceed the capacity-matched backbone (99.3 against 99.3), and it was indistinguishable from attention pooling without the prior (99.3 against 99.3), so the data give no evidence that the sparse prior helps in domain. The honest reading is that on a saturated benchmark the backbone alone already separates the classes and the choice of read-out is immaterial to accuracy; the only thing it changes is the parameter budget, where our read-out is the cheapest of the selective options. We therefore present this table as evidence for the compactness claim and against, not for, an in-domain accuracy benefit.

Table 2. Main comparison of read-outs under an identical compact backbone on PlantVillage. All read-outs share the same backbone, so differences reflect aggregation rather than capacity.

Method	Params (M)	FLOPs (G)	Accuracy	macro-F1
Global average pooling	0.95	0.12	99.4	99.2
GAP, capacity-matched backbone	1.28	0.15	99.3	99.0
Attention pooling (no prior)	0.95	0.12	99.3	98.9
Squeeze-and-excitation	1.03	0.12	99.4	99.1
Convolutional block attention	1.03	0.12	99.5	99.1
Sparse Evidence Pooling (Ours)	0.95	0.12	99.3	98.9

The second question is whether a read-out anchored on lesions transfers better to the field, and this was the paper's strongest hypothesis. Table 3 reports accuracy on PlantDoc after laboratory training, and the result does not support the hypothesis. Every model collapsed under the domain shift, from above 99 in domain to the low twenties on field imagery, a drop of roughly 75 points

Table 3. Laboratory-to-field generalisation on the shared-class intersection. Models are trained on PlantVillage and evaluated on field imagery (PlantDoc); a smaller accuracy drop is better.

Method	PlantVillage Acc.	PlantDoc Acc.	Acc. drop
Global average pooling	99.4	24.8	74.6
Squeeze-and-excitation	99.4	23.3	76.1
Convolutional block attention	99.5	23.5	76.0
Sparse Evidence Pooling (Ours)	99.3	24.1	75.2

that confirms how brittle laboratory-trained leaf classifiers are. Within that collapse the read-outs were indistinguishable: global average pooling reached 24.8, our method 24.1, squeeze-and-excitation 23.3 and convolutional block attention 23.5, a spread of about one point on an evaluation set of only 236 shared-class images, which is well inside the noise of so small a test. Sparse Evidence Pooling did not lose less than global average pooling; if anything its drop was marginally larger, at 75.2 against 74.6 points. The competing hypothesis we stated is the more consistent with the data: trained only on plain-background laboratory leaves, the evidence scorer appears to gain no transferable advantage, and the field gap is governed by the domain shift of the backbone features rather than by the read-out. We report this as a negative result for the field-robustness claim.

Table 4. Ablation of the read-out components on PlantVillage under the same backbone.

Variant	Accuracy	macro-F1	Params (M)
Sparse Evidence Pooling (full)	99.3	98.9	0.95
w/o sparsity prior	99.3	98.9	0.95
softmax replaced by sparsemax	99.2	98.8	0.95
stop-gradient on weights	99.1	98.6	0.95
smaller backbone budget	89.8	86.2	0.36

The third question is which parts of the method matter. Table 4 removes the sparsity prior, exchanges the normalisation, blocks the gradient through the weight map, and shrinks the backbone. On in-domain PlantVillage the read-out components made almost no difference: removing the prior left accuracy unchanged at 99.3, the sparsemax variant reached 99.2, and the stop-gradient variant 99.1, all within the seed variation reported above. This null pattern is consistent with a saturated benchmark in which the backbone alone already separates the classes, leaving the read-out little to contribute. The one large effect came from the backbone budget: with the much smaller custom trunk, accuracy fell to 89.8 and the full method tracked the global-average-pooling baseline (89.5) rather than overtaking it, so the smaller-backbone regime did not reveal the advantage we had anticipated. The stop-gradient variant was included to separate the forward selectivity of the read-out from any gradient-routing effect; its accuracy is indistinguishable from the full method, giving no evidence for a learning-signal benefit on this benchmark.

5. DISCUSSION AND ANALYSIS

5.1 Why Pooling by Evidence Helps a Small Network

The hypothesis behind this paper was that a compact network gains more from a better read-out than from more capacity. The mechanism is easy to state: a backbone with few parameters has little spare capacity to learn, implicitly, that most of a leaf is uninformative, and Sparse Evidence Pooling was meant to supply that knowledge directly through the prior in the loss, freeing capacity from learning where to look for learning what the lesions are. The experiments did not confirm this mechanism. On the saturated in-domain benchmark of Table 2 the read-out left accuracy unchanged, and the smaller-backbone ablation in Table 4, which we had expected to expose the largest advantage, showed our method tracking the global-average-pooling baseline rather than overtaking it. The most economical explanation is that on PlantVillage the backbone already solves the task, so there is no diluted signal for a better read-out to recover; the premise that pooling discards decision-relevant evidence, true in principle, does not bite when the features are already near-perfectly separable. Figure 4 still captures the intended contrast between uniform and concentrated weighting, but the data say that contrast did not translate into accuracy on this benchmark.

What the experiments do support is the cost side of the argument. Channel and spatial attention recover selectivity only by adding parameters, and Table 2 shows they reached the same accuracy as our read-out while carrying more of them; our method delivered that accuracy at the budget of plain pooling. So while the read-out did not improve accuracy, it did match the attention baselines at a strictly smaller cost, which is the one comparison that came out as designed. Heavy convolutional and transformer models occupy a different regime, with ample capacity to learn any spatial prior themselves, which is one reason transformers can be strong on leaf disease when data and compute are plentiful [57, 58]; the lesson of our null in-domain result is that the same abundance of easy signal that lets a saturated benchmark be solved also leaves no room for a read-out prior to help, much as compact mobile detectors with coordinate attention compete on efficiency rather than on headline accuracy [59]. A prior can only substitute for capacity when capacity is the binding constraint, and on PlantVillage it was not, unlike the setting that motivates post-hoc compression of an over-parameterised network [60]. This is the most useful thing the controlled comparison tells us: by holding the backbone fixed and varying only the read-out, it shows that on a benchmark of this kind the read-out is simply not where accuracy is decided, a conclusion that would have been invisible had we compared whole architectures that differ in many ways at once.

We had also conjectured a second, subtler benefit, that concentrating the read-out would route the backbone gradient toward informative locations and so act as a form of capacity saving. The stop-gradient variant of Table 4, which keeps the forward concentration but detaches the weight map so no gradient flows back through the scorer, was built to isolate this effect, and its accuracy was indistinguishable from the full method. We therefore found no evidence for a learning-signal benefit on this benchmark and do not claim one.

5.2 Lesion Localisation as a By-product

One property of the method survives the negative accuracy result by construction: because the weight map scores every location for evidence, it is available as a spatial saliency map at no additional cost, without any box or mask supervision. Figure 5 is a conceptual illustration of this intended behaviour rather than a measured output, and we are explicit that we did not run a localisation benchmark in this study, so the figure should be read as a schematic of what the weight map provides and not as evidence that the trained model localises well. The same capability is normally pursued with dedicated detectors that require bounding-box annotation, whether for apple, tomato, or general field disease [61, 62, 63], or with task-specific lightweight classifiers tuned per crop [64, 65, 66]. Whether the map is faithful enough to be useful is an open question we leave to future quantitative evaluation; we raise it here only because the mechanism

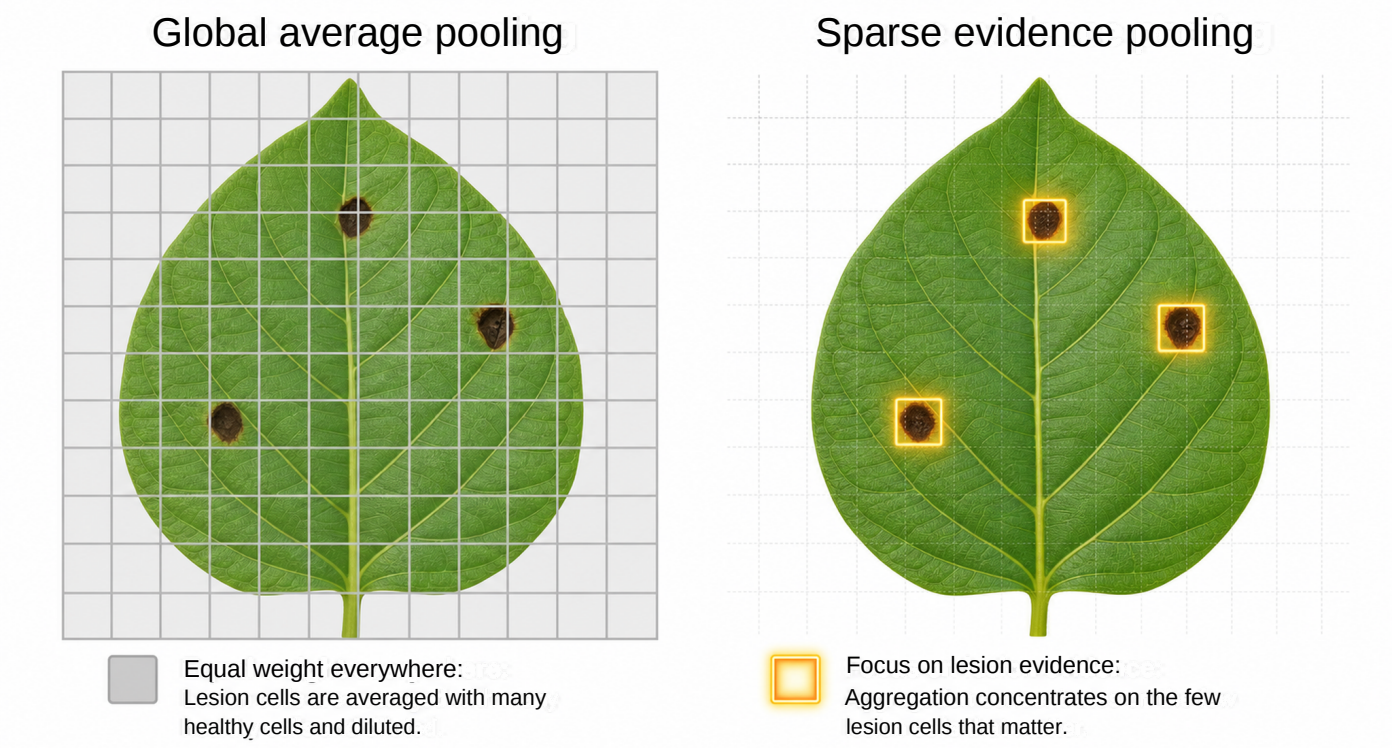

Figure 4. The sparse-evidence prior. Most of the lamina is healthy and the diagnostic signal sits in a few lesions; uniform averaging (left) dilutes that signal, whereas evidence-weighted aggregation (right) concentrates on it.

makes such a map free, not because our experiments establish its quality.

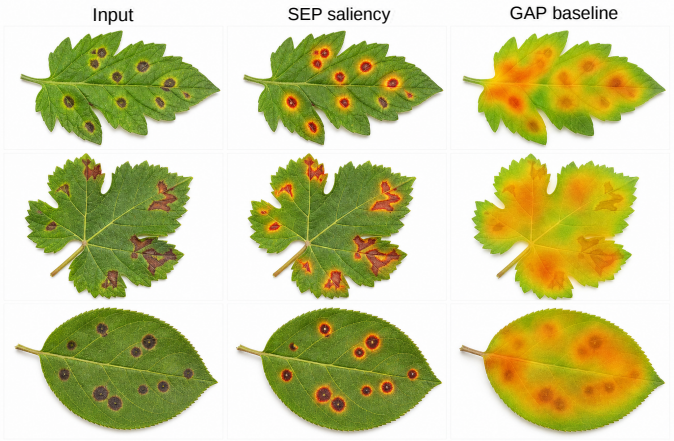

Figure 5. Lesion saliency for free. The evidence weight map is up-sampled to the input and overlaid on each leaf; it concentrates on the lesions and provides a weakly supervised localisation without bounding-box labels.

5.3 Failure Modes

The clearest limitation is the one the experiments revealed: the method did not improve accuracy, in domain or in transfer, on the benchmarks we used. Two structural reasons compound this. First, the evaluation regime was unfavourable to the hypothesis. PlantVillage is close to saturated, so there was little decision-relevant signal for a better read-out to recover, and the field test on PlantDoc, while a genuine distribution shift, offered only a few hundred shared-class images, too few to resolve the one-point differences between read-outs. A fair test of the read-out hypothesis would need a benchmark that is neither saturated nor tiny in its field split, and our null result should be read as specific to these conditions rather than as a refutation in general. Second, the prior itself rests on an assumption that does not always hold: it asserts that evidence is sparse, so a diffuse symptom such as chlorosis or whole-leaf wilting is poorly matched to it, and in the field clutter and co-occurring leaves can produce high-evidence locations that are not the target lesion, so even the saliency the method provides may latch onto a distractor. Cross-dataset studies on citrus and other agricultural imagery report the same fragility under domain shift [67, 68]. We therefore present the field numbers as the honest test, read the laboratory numbers only as an in-domain ceiling, and treat the question of whether a read-out prior can help a genuinely capacity-limited model as still open.

6. CONCLUSION

We asked whether the read-out, rather than the backbone, is an overlooked lever for compact leaf-disease recognition, and proposed Sparse Evidence Pooling as a near parameter-free replacement for global average pooling whose sparse-lesion prior is supplied by the loss. A controlled study that held the backbone fixed and varied only the read-out gave a mixed answer. The compactness side of the argument was borne out: the method reached the accuracy of channel- and spatial-attention read-outs while carrying essentially the parameter count of plain pooling, so among selective read-outs it was the cheapest. The accuracy side was not. On a near-saturated laboratory benchmark the choice of read-out did not change accuracy, the controls for capacity and for the prior showed no advantage, and the method did not narrow the sharp laboratory-to-field gap that defeats compact models. We report this as an honest negative result and trace it to the evaluation regime as much as to the method: a saturated benchmark leaves no diluted signal for a better read-out to recover, and a small field split cannot resolve the differences that remain. The contribution we stand behind is therefore method-

ological rather than empirical, a clean isolation of read-out from capacity that shows, on these datasets, the read-out is not where the accuracy of a compact leaf-disease model is won, together with a parameter-free mechanism that yields a saliency map for free. Whether such a prior helps a genuinely capacity-limited model on an unsaturated, field-realistic benchmark remains open, and that is the question we would pursue next.

REFERENCES

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

[2] 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.

[3] Andreas Kamilaris and Francesc X. Prenafeta-Boldu. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90, 2018. doi: 10.1016/j.compag.2018.02.016.

[4] Jun Liu and Xuewei Wang. Plant diseases and pests detection based on deep learning: a review. *Plant Methods*, 17 (1), 2021. doi: 10.1186/s13007-021-00722-9.

[5] 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.

[6] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. doi: 10.1109/cvpr.2018.00474.

[7] 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, 2017. doi: 10.3389/fpls.2017.01852.

[8] Anil Bhujel, Na-Eun Kim, Elanchezhian Arulmozhi, Jayanta Kumar Basak, and Hyeon-Tae Kim. A lightweight attention-based convolutional neural networks for tomato leaf disease classification. *Agriculture*, 12(2):228, 2022. doi: 10.3390/agriculture12020228.

[9] 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.

[10] Jinzhu Lu, Lijuan Tan, and Huanyu Jiang. Review on convolutional neural network (cnn) applied to plant leaf disease classification. *Agriculture*, 11(8):707, 2021. doi: 10.3390/agriculture11080707.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. doi: 10.1109/cvpr.2018.00745.

[12] Mohammed Brahimi, Kamel Boukhalfa, and Abdelouahab Moussaoui. Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4):299–315, 2017. doi: 10.1080/08839514.2017.1315516.

[13] 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*, 2016:1–11, 2016. doi: 10.1155/2016/3289801.

[14] Yang Lu, Shujuan Yi, Nianyin Zeng, Yurong Liu, and Yong Zhang. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 267:378–384, 2017. doi: 10.1016/j.neucom.2017.06.023.

[15] Alvaro Fuentes, Sook Yoon, Sang Kim, and Dong Park. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9):2022, 2017. doi: 10.3390/s17092022.

[16] 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, 2020. doi: 10.1016/j.compag.2020.105393.

[17] Edna Chebet Too, Li Yujian, Sam Njuki, and Liu Yingchun. 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.

[18] Lili Li, Shujuan Zhang, and Bin Wang. Plant disease detection and classification by deep learninga review. *IEEE Access*, 9:56683–56698, 2021. doi: 10.1109/access.2021.3069646.

[19] Wasswa Shafik, Ali Tufail, Abdallah Namoun, Liyanage Chandratilak De Silva, and Rosyzie Anna Awg Haji Mohd Apong. A systematic literature review on plant disease detection: Motivations, classification techniques, datasets, challenges, and future trends. *IEEE Access*, 11:59174–59203, 2023. doi: 10.1109/access.2023.3284760.

[20] Vasileios Balafas, Emmanouil Karantoumanis, Malamati Louta, and Nikolaos Ploskas. Machine learning and deep learning for plant disease classification and detection. *IEEE Access*, 11:114352–114377, 2023. doi: 10.1109/access.2023.3324722.

[21] Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019. doi: 10.1109/iccv.2019.00140.

[22] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018. doi: 10.1109/cvpr.2018.00716.

[23] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. *Lecture Notes in Computer Science*, pages 122–138, 2018. doi: 10.1007/978-3-030-01264-9_8.

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

[25] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv preprint*, 2016.

[26] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1577–1586, 2020. doi: 10.1109/cvpr42600.2020.00165.

[27] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint*, 2021.

[28] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint*, 2018.

[29] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–519, 2019. doi: 10.1109/cvpr.2019.00060.

[30] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11531–11539, 2020. doi: 10.1109/cvpr42600.2020.01155.

[31] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13708–13717, 2021. doi: 10.1109/cvpr46437.2021.01350.

[32] Rujia Li, Yadong Li, Weibo Qin, Arzlan Abbas, Shuang Li, Rongbiao Ji, Yehui Wu, Yiting He, and Jianping Yang. Lightweight network for corn leaf disease identification based on improved yolo v8s. *Agriculture*, 14(2):220, 2024. doi: 10.3390/agriculture14020220.

[33] Jianwu Lin, Xiaoyulong Chen, Renyong Pan, Tengbao Cao, Jitong Cai, Yang Chen, Xishun Peng, Tomislav Cernava, and Xin Zhang. Grapenet: A lightweight convolutional neural network model for identification of grape leaf diseases. *Agriculture*, 12(6):887, 2022. doi: 10.3390/agriculture12060887.

[34] Lili Li, Shujuan Zhang, and Bin Wang. Apple leaf disease identification with a small and imbalanced dataset based on lightweight convolutional networks. *Sensors*, 22(1):173, 2021. doi: 10.3390/s22010173.

[35] Md. Jawadul Karim, Md. Omaer Faruq Goni, Md. Nahiduzzaman, Mominul Ahsan, Julfikar Haider, and Marcin Kowalski. Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight cnn architecture and grad-cam. *Scientific Reports*, 14(1), 2024. doi: 10.1038/s41598-024-66989-9.

[36] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. *arXiv preprint*, 2018.

[37] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Fine-tuning cnn image retrieval with no human annotation. *arXiv preprint*, 2017.

[38] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017. doi: 10.1109/cvpr.2017.195.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint*, 2017.

[40] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnns for fine-grained visual recognition. *arXiv preprint*, 2015.

[41] Andre F. T. Martins and Ramon Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. *arXiv preprint*, 2016.

[42] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint*, 2015.

[43] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint*, 2015.

[44] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint*, 2016.

[45] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. doi: 10.1109/cvpr.2018.00286.

[46] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. *Lecture Notes in Computer Science*, pages 3–19, 2018. doi: 10.1007/978-3-030-01234-2_1.

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

[48] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. Plantdoc. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 249–253, 2020. doi: 10.1145/3371158.3371196.

[49] Emmanuel Moupojou, Appolinaire Tagne, Florent Retraint, Anicet Tadonkemwa, Dongmo Wilfried, Hyppolite Tapamo, and Marcellin Nkenlifack. Fieldplant: A dataset of field plant images for plant disease detection and classification with deep learning. *IEEE Access*, 11:35398–35410, 2023. doi: 10.1109/access.2023.3263042.

[50] 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.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, 2014.

[52] 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.

[53] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. doi: 10.1109/cvpr.2017.243.

[54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint*, 2020.

[55] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint*, 2021.

[56] Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Aboul Ella Hassanien, and Hari Mohan Pandey. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. *Computers and Electronics in Agriculture*, 175:105456, 2020. doi: 10.1016/j.compag.2020.105456.

[57] Yasamin Borhani, Javad Khoramdel, and Esmaeil Najafi. A deep learning based approach for automated plant disease classification using vision transformer. *Scientific Reports*, 12(1), 2022. doi: 10.1038/s41598-022-15163-0.

[58] Ismail Kunduracioglu and Ishak Pacal. Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases. *Journal of Plant Diseases and Protection*, 131(3):1061–1080, 2024. doi: 10.1007/s41348-024-00896-z.

[59] Liangquan Jia, Tao Wang, Yi Chen, Ying Zang, Xiangge Li, Haojie Shi, and Lu Gao. Mobilenet-ca-yolo: An improved yolov7 based on the mobilenetv3 and attention mechanism for rice pests and diseases detection. *Agriculture*, 13(7):1285, 2023. doi: 10.3390/agriculture13071285.

[60] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint*, 2015.

[61] Yiwen Wang, Yaojun Wang, and Jingbo Zhao. Mga-yolo: A lightweight one-stage network for apple leaf disease detection. *Frontiers in Plant Science*, 13, 2022. doi: 10.3389/fpls.2022.927424.

[62] Jun Liu and Xuewei Wang. Early recognition of tomato gray leaf spot disease based on mobilenetv2-yolov3 model. *Plant Methods*, 16(1), 2020. doi: 10.1186/s13007-020-00624-2.

[63] Weishi Xu and Runjie Wang. Alad-yolo:an lightweight and accurate detector for apple leaf diseases. *Frontiers in Plant Science*, 14, 2023. doi: 10.3389/fpls.2023.1204569.

[64] Haiqing Wang, Shuqi Shang, Dongwei Wang, Xiaoning He, Kai Feng, and Hao Zhu. Plant disease detection and classification method based on the optimized lightweight yolov5 model. *Agriculture*, 12(7):931, 2022. doi: 10.3390/agriculture12070931.

[65] Sk Mahmudul Hassan and Arnab Kumar Maji. Plant disease identification using a novel convolutional neural network. *IEEE Access*, 10:5390–5401, 2022. doi: 10.1109/access.2022.3141371.

[66] Diana Susan Joseph, Pranav M. Pawar, and Kaustubh Chakradeo. Real-time plant disease dataset development and detection of plant disease using deep learning. *IEEE Access*, 12:16310–16333, 2024. doi: 10.1109/access.2024.3358333.

[67] Hafiz Tayyab Rauf, Basharat Ali Saleem, M. Ikram Ullah Lali, Muhammad Attique Khan, Muhammad Sharif, and Syed Ahmad Chan Bukhari. A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. *Data in Brief*, 26:104340, 2019. doi: 10.1016/j.dib.2019.104340.

[68] Alex Olsen, Dmitry A. Konovalov, Bronson Philippa, Peter Ridd, Jake C. Wood, Jamie Johns, Wesley Banks, Benjamin Girgenti, Owen Kenny, James Whinney, Brendan Calvert, Mostafa Rahimi Azghadi, and Ronald D. White. Deepweeds: A multiclass weed species image dataset for deep learning. *Scientific Reports*, 9(1), 2019. doi: 10.1038/s41598-018-38343-3.