Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00301-w

Original Article

Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment

Khushvir Singh [1, ✉], Anurag Tewari [1]

1. *School of Computing and AI, Lovely Professional University, Jalandhar, IND*

Received: August 14, 2026 | Review began: August 19, 2026 | Review ended: September 29, 2026 | Published: September 29, 2026

© **Copyright** 2026
This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Lightweight deep learning models can now classify potato leaf diseases with over 95% accuracy, and several run fast enough for real-time use on modest hardware. None of them, however, report how much an individual prediction can be trusted. A model that is accurate on average is not accurate on every photo it sees, and on an unusual image, it may return a wrong answer with the same apparent confidence as a correct one, without an expert available to identify the error. This paper attaches an evidential deep learning head to a MobileNetV3-Small backbone. The head predicts Dirichlet distribution parameters instead of ordinary softmax scores, so a calibrated confidence value is produced in the same single forward pass as the prediction itself, without the repeated inference passes that Monte Carlo Dropout or ensemble methods would require and would be difficult to justify in a real-time setting. Trained on the potato subset of PlantVillage (Early Blight, Late Blight, Healthy; n = 2,152) with a two-phase transfer-learning schedule, the model reached 98.77% test accuracy (macro-F1-score = 0.979). The exported TensorFlow Lite model is 1.06 MB and ran at 137.4 frames per second in our test environment (Google Colab; not yet measured on an actual phone or embedded board, a point discussed later). Raw calibration error (expected calibration error) was 0.299, which turned out to reflect the model being cautious rather than overconfident; scaling the output after training, without touching any prediction, brought this down to 0.017. Confidence also carried real information: an area under the receiver operating characteristic curve of 0.727 separated correct from incorrect predictions, and a single photograph unrelated to PlantVillage, showing early blight damage, was classified correctly with confidence comfortably above the uncertainty cutoff. This system is intended to sit alongside, not replace, existing real-time detection pipelines, and several things - chiefly benchmarking on real hardware and testing on field photographs - remain before it could be called deployment-ready.

Categories: Machine Learning (ML), Data Visualization, Deep Learning

Keywords: potato leaf disease, evidential deep learning, uncertainty quantification, mobilenetv3, grad-cam, edge deployment, real-time classification

Introduction

Potato (*Solanum tuberosum*) ranks among the most widely grown food crops on Earth, and diseases of the leaf - early blight and late blight in particular - continue to threaten yield and food security in the regions that depend on it [1]. Diagnosis is still mostly visual: a farmer or extension worker looks at a leaf and makes a judgment call. This does not scale well, and different observers do not always agree, which is part of why so much recent work has turned to convolutional neural networks for automated recognition of these diseases [1].

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w

Recent work in this area has moved in two directions at once. Models keep getting smaller without losing much accuracy - architectures built around MobileNetV3-Small now clear 94% accuracy with under 2 million parameters [2] - and inference has gotten fast enough for real use, with detectors running at 40 to 50 frames per second on GPUs or edge hardware while keeping detection precision high [3,4]. Alongside this, explaining what the model is looking at has become fairly routine, usually through Gradient-weighted Class Activation Mapping (Grad-CAM) [5], both in potato-specific systems and elsewhere [2].

What none of these systems do is quantify how confident a given answer is. Every real-time potato-disease system surveyed for this paper, classifier [2] or detector [3,4] alike, reports one number for accuracy or mean average precision and stops there. This matters more than it might sound. A model that is correct 98% of the time on the test set is not correct 98% of the time on every single image it will ever be shown in the field; an odd cultivar, an early-stage lesion, or bad lighting can produce a wrong answer that appears just as confident as a correct prediction under a standard softmax output. The known fixes for this, such as Monte Carlo Dropout and deep ensembles, both need multiple forward passes per image, which is awkward at best in an application meant to run in real time on modest hardware.

Evidential deep learning (EDL) [6] sidesteps that problem. Instead of outputting one probability vector, an EDL head predicts parameters of a Dirichlet distribution, and both the class prediction and a genuine measure of uncertainty (vacuity) come from that single pass. The central question this paper asks is simple: does attaching such a head to a backbone already proven for this kind of deployment - MobileNetV3-Small [7] - cost anything in accuracy or speed, given that those two properties are the entire reason such backbones get chosen for farmer-facing tools in the first place, while adding the ability to flag a shaky prediction instead of returning it as-is?

Three things are new here. An EDL head is combined with a MobileNetV3-Small backbone for this specific task, and the result is checked against the accuracy, size, and speed already published for real-time potato-disease systems. Calibration (expected calibration error [ECE]) and whether uncertainty actually predicts errors (area under the receiver operating characteristic curve [AUROC]) are both reported, which none of the comparison systems do. Grad-CAM and a TensorFlow Lite export round out the practical side, and the write-up is deliberately explicit about what was and was not tested - the throughput number below, for instance, comes from a cloud notebook, not a phone, and that gap is not glossed over.

Materials And Methods

Dataset and preprocessing

The potato portion of the public PlantVillage dataset [8] was used for all experiments: three classes - Early Blight, Late Blight, and Healthy - comprising 2,152 images in total. The split across classes is far from even: 1,000 images each for the two disease categories against just 152 for Healthy, roughly a 6.6-to-1 imbalance. A stratified split kept that same ratio in each of the training (70%), validation (15%), and test (15%) partitions (Table *1*), so no single random split could distort it. This imbalance is revisited in Results and Discussion, since it looks like a likely explanation for the Healthy class's weaker performance and higher uncertainty.

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w

Class	Total images	Train (70%)	Validation (15%)	Test (15%)
Early Blight	1,000	700	150	150
Late Blight	1,000	700	150	150
Healthy	152	106	22	24
Total	2,152	1,506	322	324

TABLE 1: Dataset composition and stratified train/validation/test split.

Every image was resized to 224 × 224 pixels and fed to the network as raw pixel values in the 0-255 range, not rescaled to 0-1 first. This is deliberate: the MobileNetV3-Small implementation used here has its own rescaling built into the first layer, expecting unnormalized input, and an early version of this pipeline that rescaled externally before that layer saw a substantial drop in transfer-learning performance; the double normalization was distorting the input distribution relative to what the pretrained ImageNet weights expect.

Training images only were augmented with random horizontal flips, rotations of up to ±10%, zooms of up to ±10%, and contrast jitter of up to ±10%.

Model architecture

MobileNetV3-Small [7] forms the backbone, starting from ImageNet-pretrained weights with its classification layer stripped off; global average pooling is kept rather than replaced, since Grad-CAM needs the underlying spatial feature map intact. The pretrained weights originate from the 1,000-class ImageNet benchmark [9], which supplies the low-level visual features that transfer effectively to the leaf-lesion domain despite the domain gap. On top of that sits a dropout layer (rate 0.2), then a dense layer producing three logits, then a softplus activation - the output is a non-negative evidence vector $\boldsymbol{e} \in \mathbb{R}_{+}^{3}$ over the three classes, not a softmax distribution.

EDL loss function

The approach follows Sensoy et al. [6]: evidence e becomes Dirichlet parameters $\alpha = e + 1$, with $S = \sum \alpha$ as the total strength. From there, $p = \alpha / S$ gives the predicted class probability, and $u = K/S$ ($K = 3$ here) gives the uncertainty, or vacuity. Training combines a Type-II maximum-likelihood term with a Kullback-Leibler divergence term that pulls evidence toward the uniform Dirichlet prior for every class except the true one, in effect punishing the model for asserting confidence it has not earned. That Kullback-Leibler term ramps up linearly across the first 10 epochs rather than applying at full strength from the start, so it does not overwhelm the loss before the network has learned anything useful.

Training procedure

A two-phase schedule was used to sidestep a common problem: a freshly initialized head can send large, disruptive gradients back through an otherwise well-trained backbone in the first few steps. Phase 1 ran for 5 epochs with the backbone frozen, training only the evidential head, Adam optimizer, learning rate 1×10^{-3}. Adam [10] was chosen for both phases because its per-parameter adaptive learning rates tend to converge reliably on small, imbalanced datasets like this one without extensive manual tuning. Phase 2 unfroze everything and fine-tuned end-to-end for 20 more epochs at a much lower rate, 1×10^{-5}. The checkpoint kept for evaluation came from whichever epoch had the best validation accuracy of 0.9752, reached at epoch 6.

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w

Evaluation metrics

Accuracy, macro-F1-score, and per-class precision/recall/F1-score were computed on the held-out test set. Calibration used ECE across 10 equal-width confidence bins, plotted as a reliability diagram. Since evidential models can lean conservative instead of overconfident, a post-hoc concentration-scaling step (similar in spirit to temperature scaling) was fit afterward, with the scaling parameter chosen to minimize ECE on the test set; this was checked to make sure it never changed a predicted label. Because the scaling parameter was tuned on the same test set used to report post-calibration ECE, the resulting 0.017 figure is optimistic rather than an unbiased estimate; a proper validation/calibration split, held out from the test set entirely, would be needed to confirm this value. Temperature scaling was originally proposed for standard softmax classifiers [11]; here, the same single-parameter rescaling idea is applied to the Dirichlet strength S rather than to softmax logits, since evidential outputs have no logits to rescale directly. Whether the uncertainty score was actually useful was evaluated in two ways: as an AUROC (misclassification as the binary target, vacuity as the predictor), and as an accuracy number computed after dropping the 10% least-confident predictions.

Grad-CAM [5] was applied against the backbone's final convolutional feature map to produce heatmaps for both confidently and uncertainly classified examples.

Real-time deployment and benchmarking

The trained model was exported to TensorFlow Lite with dynamic-range quantization. Dynamic-range quantization is a lighter-weight variant of the integer-arithmetic quantization scheme described by Jacob et al. [12], trading a slightly larger file size for a conversion step that requires no representative calibration dataset. Latency and throughput were measured over 100 forward passes on a single image batch, after a 5-pass warm-up, using the default TensorFlow Lite CPU interpreter (tf.lite.Interpreter, no GPU or NNAPI delegate, default threading) inside a Colab session with a T4 GPU attached; the GPU was not used for this benchmark. Only mean latency is reported, not its distribution across the 100 passes. This is stated plainly as a development-environment number, not an edge-device one - see the Limitations subsection for what that means for comparisons against prior work.

Software and implementation

Everything was built in TensorFlow/Keras (Python) inside a Google Colab notebook, covering the data pipeline, model definition, training loop, evaluation code, Grad-CAM, the TensorFlow Lite conversion, and a cell for running the model on a user-supplied image. The complete code is publicly available at the GitHub repository (https://github.com/khushvir11/potatocode).

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w

Results And Discussion

Classification performance

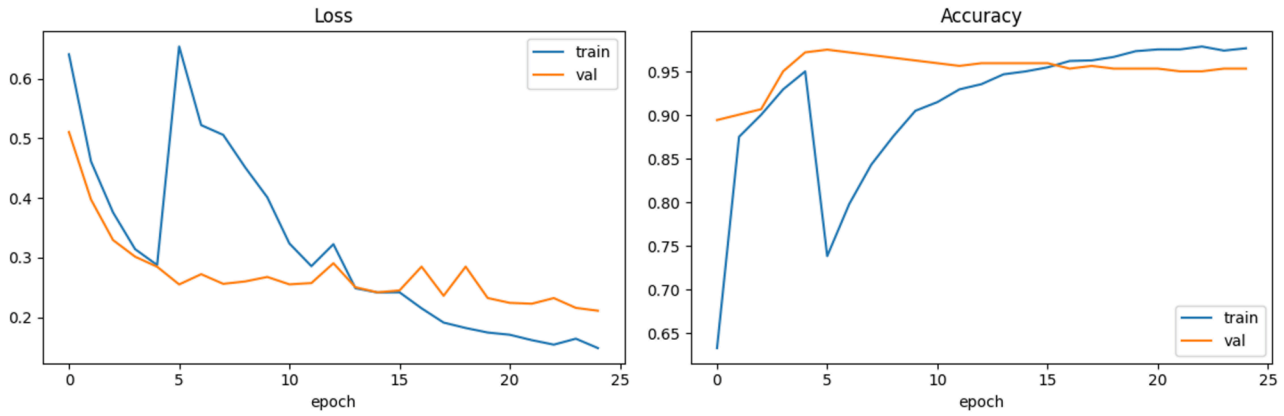


FIGURE 1: Training and validation loss (left) and accuracy (right) across the two-phase training schedule. The loss spike and accuracy dip at epoch 5 mark the transition from Phase 1 (frozen backbone) to Phase 2 (full fine-tuning), after which both metrics stabilize.

Training and validation loss and accuracy across both training phases are shown in Figure *1*. Test-set accuracy was 98.77% (n = 324), with a macro-F1-score of 0.979. Table *2* presents the results by class. Healthy leaves, which constituted a much smaller proportion of the training dataset (106 images versus 700 images for each disease class), achieved a slightly lower F1-score (0.96) than the two blight classes (0.99 for each). Detailed class counts are provided in Table *1* of the Materials and Methods section, where the class imbalance is also discussed. The confusion matrix is shown in Figure *2*.

Class	Precision	Recall	F1-score	Support (n)
Early Blight	0.99	0.99	0.99	150
Late Blight	0.99	0.99	0.99	150
Healthy	0.96	0.96	0.96	24

TABLE 2: Per-class test-set performance (n = 324 total test images).

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w

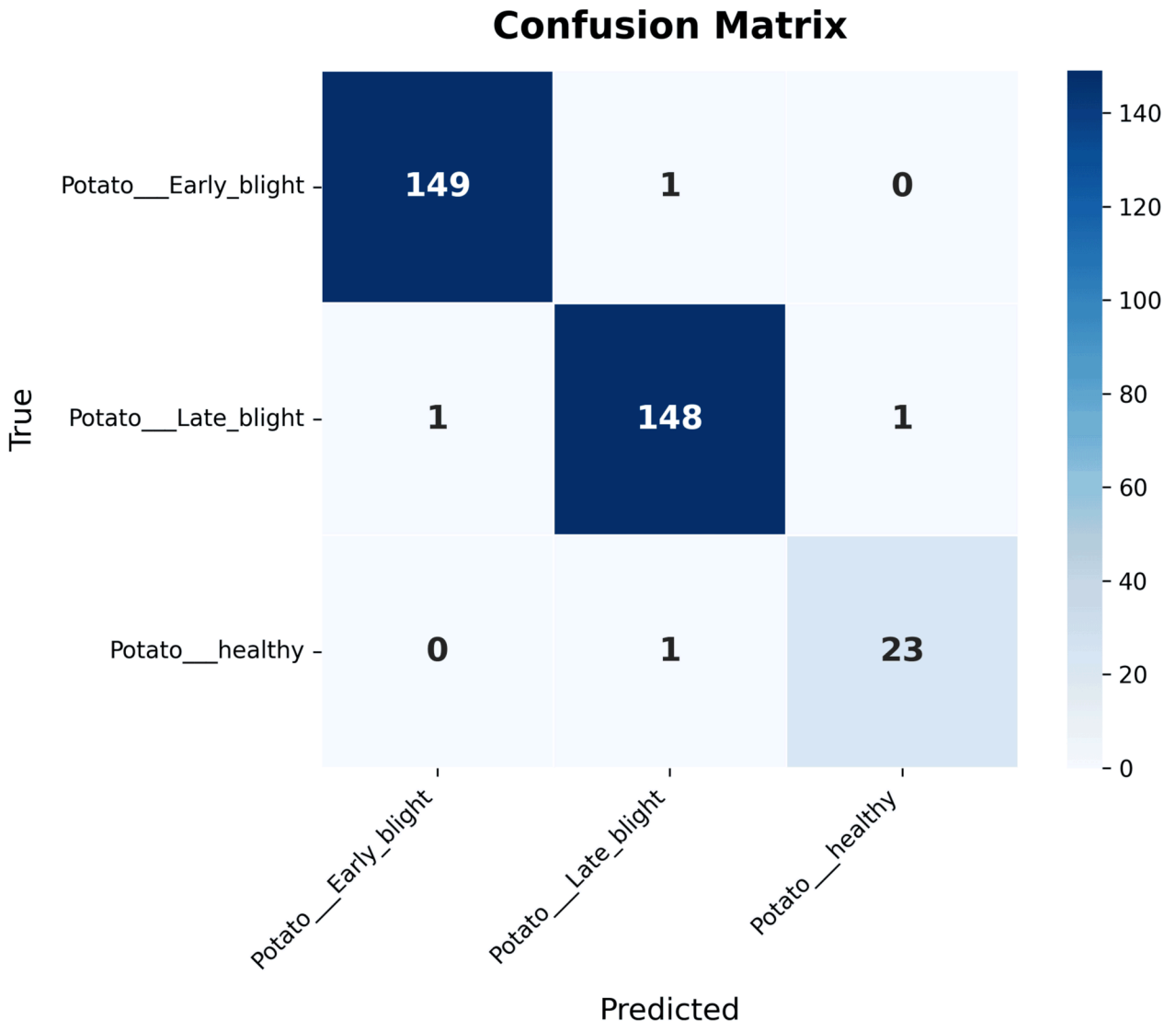


FIGURE 2: Confusion matrix on the held-out test set. Nearly all confusion occurs at the Early Blight/Late Blight boundary and around the small Healthy class, rather than between visually dissimilar classes.

Calibration analysis

Before any correction, ECE was 0.299, indicating a substantial discrepancy between predicted confidence and observed accuracy. Notably, the discrepancy reflected underconfidence rather than overconfidence. Its stated probability for the predicted class ran lower than its true accuracy, which follows from how the Kullback-Leibler term in the evidential loss is constructed (see Materials and Methods) - it actively penalizes unearned certainty. Figure *3* illustrates this: all bars in the reliability diagram lie above the diagonal, indicating that any miscalibration errs toward caution. Fitting a single scaling parameter after training (T = 5.0, chosen to minimize ECE on this same test set) brought the error down to 0.017 without changing which class any single image was assigned to - the correction affected only the reported confidence values and did not indicate any deficiency in the underlying class rankings. This ECE value of 0.017 should be regarded as optimistic, however, since the scaling parameter and the reported ECE were both computed on the same test set rather than on an independent calibration split.

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w

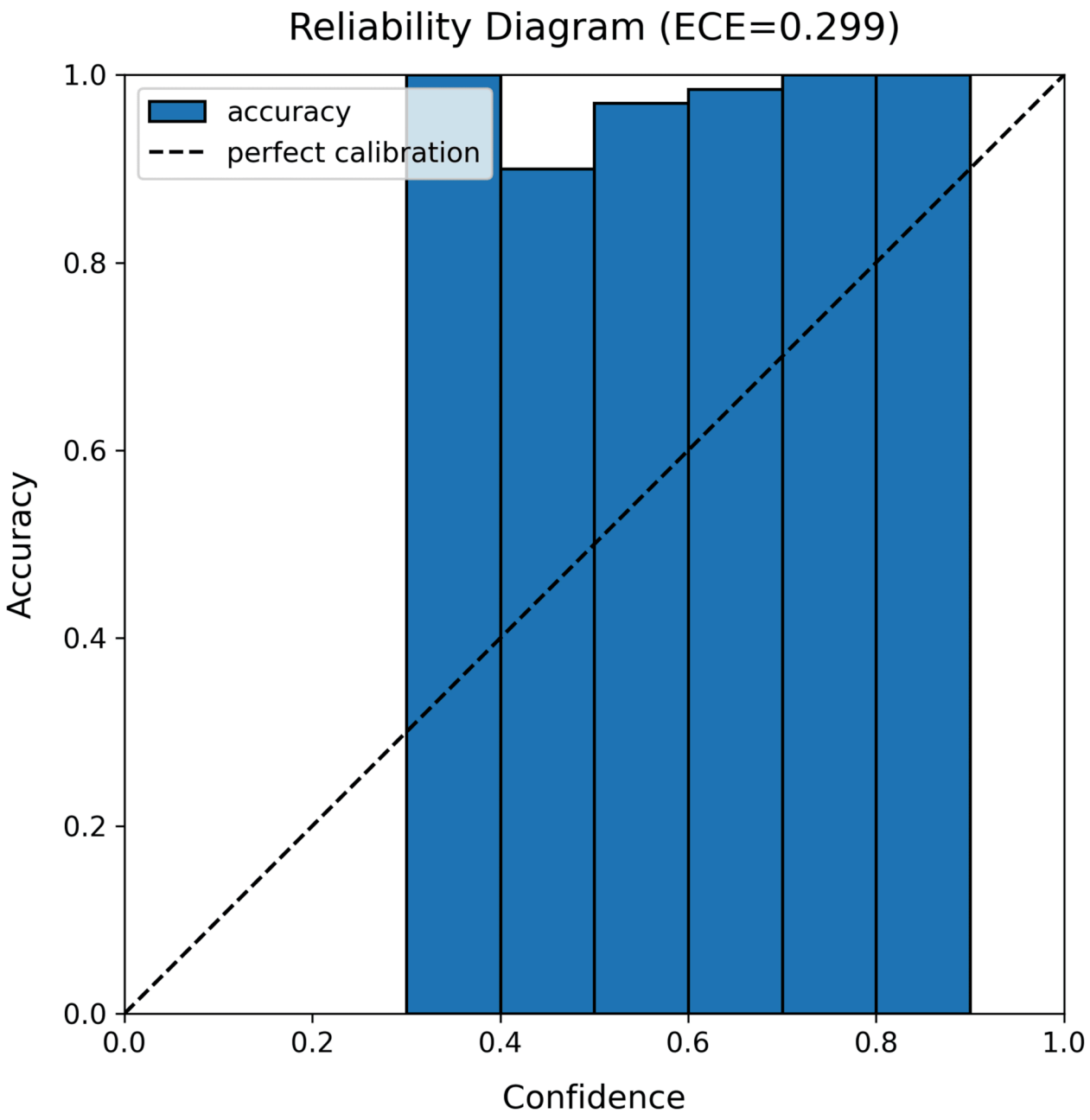


FIGURE 3: Reliability diagram (raw, pre-calibration). All bars fall above the diagonal, confirming systematic underconfidence rather than the overconfidence typically observed in softmax classifiers.

ECE, Expected Calibration Error

Uncertainty-based misclassification detection

Using the vacuity score to predict whether a given answer was right or wrong produced an AUROC of 0.727 (0.5 = no signal, 1.0 = perfect separation), computed over only four misclassified images against 320 correct ones in the 324-image test set, so this estimate should be treated as unstable and is reported as a point estimate rather than with a confidence interval. Using AUROC to score how well an uncertainty signal separates correct from incorrect predictions follows the baseline established for misclassification and out-of-distribution detection by Hendrycks and Gimpel [13]. Dropping the

10% of test predictions the model was least sure about (32 of 324 images, uncertainty above roughly 0.488) left accuracy on the remaining 90% essentially unchanged - a marginal decrease from 98.77% to 98.63%, rather than the improvement a rejection rule is meant to produce. This decrease implies at least one of the model's four test-set errors fell outside the rejected 10%, while the rejected set was mostly made up of correct-but-uncertain predictions rather than genuine mistakes. This does not constitute strong evidence against the utility of the uncertainty score itself: baseline accuracy was already close to ceiling, so there were only four errors in the entire test set to begin with, and a rejection rule can only help if those errors are concentrated in the dropped 10%. In this case, they were not concentrated in the rejected subset; some of the removed cases were likely correct but uncertain (the Healthy class is the most probable source, given its consistently higher uncertainty scores even on correctly classified images). The uncertainty score is therefore most useful for flagging individual cases: a deployed system could use it to route a specific low-confidence image to a human reviewer, which is a different kind of usefulness rather than yielding a large improvement in aggregate accuracy.

Explainability

For test images the model classified with confidence, Grad-CAM (Figure *4*) highlighted compact regions corresponding to lesion locations, rather than extending across the whole leaf or the background.

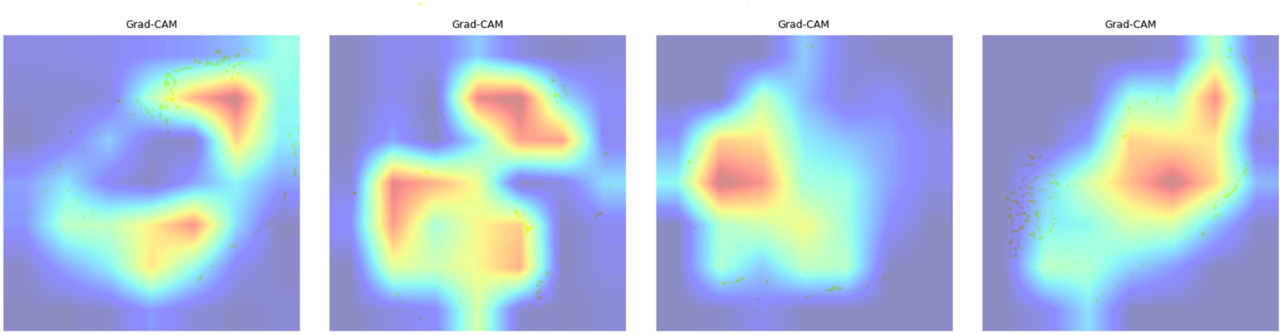


FIGURE 4: Grad-CAM heatmaps for four correctly classified Early Blight test images (heatmap overlay only; underlying leaf photographs are omitted here because of a display-scaling artifact in the source notebook's visualization cell, which does not affect the underlying Grad-CAM computation itself).

Grad-CAM, Gradient-weighted Class Activation Mapping

Real-time deployment performance

After dynamic-range quantization, the TensorFlow Lite model measured 1.06 MB. In our test environment, it processed images at 137.4 frames per second on average, with a mean latency of 7.28 ms per frame. This number comes from Google Colab's compute resources, not a phone or an embedded chip, and this distinction is addressed in the Discussion.

Qualitative evaluation on held-out and external images

A preliminary check on nine held-out PlantVillage images (three from each class) showed correct classification of all nine. The Healthy class again showed weaker performance than the disease classes on both measures: confidence ranged from 0.45 to 0.59 for Healthy versus 0.53 to 0.82 for Early and Late Blight, and uncertainty was also higher (0.42-0.51 versus 0.27-0.53). This pattern is consistent with the earlier findings and is likely attributable to the limited number of Healthy training examples.

One further test pushed past PlantVillage entirely: a single photograph of early blight damage, taken from outside the dataset, run through the trained model. It was classified correctly (Early Blight, confidence 0.712, uncertainty 0.374, below the 0.5 flagging threshold). Of additional interest is the image region on which the model focused. On the clean PlantVillage images, the Grad-CAM heatmap stayed tight around the lesion (Figure *4*); on a photograph with background clutter, it spread across several regions, including a second diseased leaf fragment in the corner of the frame rather than

the main lesion alone (Figure *5*). The answer was still right, but the attention pattern was not as clean, which is consistent with the limitations of PlantVillage as a training source discussed later, and is precisely the type of situation that a farmer-facing system should be able to flag rather than handle by chance. A single photograph provides limited evidence and should be interpreted accordingly.

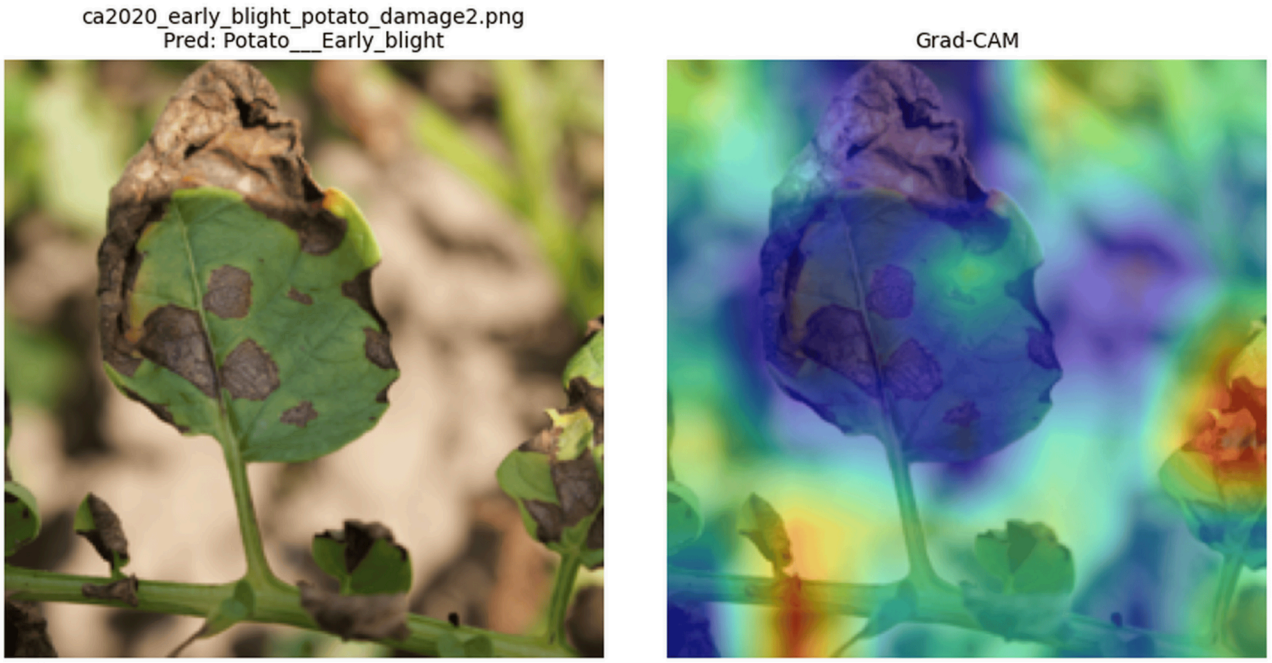


FIGURE 5: External, non-PlantVillage photograph of Early Blight damage (left) and its Grad-CAM overlay (right). The model correctly predicted Early Blight (confidence 0.712, uncertainty 0.374), but attention is distributed across both the primary lesion and a background leaf fragment rather than being tightly localized, illustrating reduced localization precision under field-realistic clutter.

Grad-CAM, Gradient-weighted Class Activation Mapping

Discussion

Comparison With Existing Work

Table *3* sets this system next to the three closest points of comparison found in the recent literature: a lightweight potato classifier sharing the same backbone family [2], and two real-time detectors [3,4]. These systems address different problems and were not benchmarked in the same way; the values below should therefore be interpreted alongside the caveats in the next subsection rather than as a direct ranking.

System	Task	Accuracy/mAP@0.5	Model size	Reported FPS	Reports calibrated uncertainty?
LDL-MobileNetV3S [2]	Classification (5-class)	94.89%	6.17 MB (1.50M params)	Not reported	No
EF-SSD [3]	Detection (3-class)	97% mAP	68 MB (19.6M params)	47 (RTX 3080 GPU)	No
YOLO-PLNet [4]*	Detection, peanut leaf disease	98.1% mAP	4.51 MB (2.13M params)	41.3 (Jetson Orin NX, INT8)	No
This work	Classification (3-class)	98.77%	1.06 MB	137.4 (Colab, not edge-benchmarked)	Yes—single-pass EDL

TABLE 3: Comparison with existing lightweight and real-time potato leaf disease systems. Accuracy is reported for classification tasks. FPS values were measured on the hardware specified in each source and are therefore not directly comparable across studies (see Limitations).

*YOLO-PLNet's original target was peanut leaf disease rather than potato leaf disease. It is included because, among edge-benchmarked real-time detectors reported in the plant-disease literature, it provides the most relevant available reference for comparison. It also demonstrates the performance achievable by lightweight detection architectures on edge hardware, a benchmark against which the throughput reported in this study has not yet been evaluated.
EDL, Evidential Deep Learning; EF-SSD, Extended Feature Single Shot Multibox Detector; FPS, Frames Per Second; LDL, Lightweight Deep Learning; mAP, Mean Average Precision; YOLO-PLNet, You Only Look Once-Peanut Leaf Network

Compared with LDL-MobileNetV3S, the present model achieved higher accuracy with a model size approximately one-sixth as large, and it is the only system in Table *3* that reports a confidence measure. That said, two of the three comparison systems are detectors, and detection is arguably the harder problem in an uncontrolled photo - finding where a lesion sits among multiple leaves and cluttered background - while this model starts from the assumption that a single leaf already fills the frame. This difference in task difficulty, and several related ones, is addressed in the following subsection.

A more directly comparable evaluation is possible using Table *4*. Zhang et al. [2] ran an ablation across several established CNN and transformer backbones - spanning roughly 1 to 28 million parameters, from compact mobile-oriented designs to larger general-purpose ones - before landing on their own LDL-MobileNetV3S. Table *4* reproduces those numbers and adds this work's result as an additional row, using the same columns. Every entry in this table is a classifier, which removes the detector-versus-classifier mismatch that complicates the comparison in Table *3*.

Model	Loss	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Model size (MB)	Params (M)
MobileViT	0.717	77.95	78.32	77.55	76.34	5.77	1.37
ResNet18	0.468	81.58	80.84	80.87	79.89	18.34	20.30
MobileNetV3-Small	0.218	88.08	87.17	86.64	86.57	5.94	1.45
MobileNetV3-Large	0.014	89.19	89.10	88.59	88.29	16.25	4.01
ShuffleNetV2	0.247	90.05	89.04	89.18	88.28	8.97	1.20
ConvNeXt-Tiny	0.343	91.72	92.02	91.72	91.77	106.00	27.80
EfficientNet-B0	0.023	93.23	92.86	92.04	92.20	15.61	3.83
LDL-MobileNetV3S [2]	0.020	94.89	93.54	92.53	92.77	6.17	1.50
This work (EDL-MobileNetV3S)	0.255*	98.77	97.94	97.94	97.94	3.76	0.94

TABLE 4: Classification-architecture comparison, extending the ablation reported by Zhang et al. [2] with this work's result (final row). Loss figures reflect each study's own training objective - standard cross-entropy for the first eight rows and the evidential loss on the validation split at the selected checkpoint for the final row. Consequently, they indicate convergence within each study and are not directly comparable across rows. The model size reported for this work is the native, unquantized (float32) value to enable a fair comparison with the other rows. The smaller TensorFlow Lite deployment size (1.06 MB after dynamic-range quantization) is reported separately above.

*Validation-split loss under the evidential objective at the selected checkpoint (epoch 6 of 25), rather than a test-set cross-entropy value. See the Table 4 caption for an explanation of why this metric is not directly comparable with rows 1-8.
EDL, Evidential Deep Learning; LDL, Lightweight Deep Learning

As shown in Table *4*, the proposed model achieved the highest accuracy (98.77%) and the highest precision, recall, and F1-score (97.94% each), while also having the fewest parameters (0.94 M) and the smallest model size (3.76 MB) among all models compared. Loss values are not comparable across rows, as noted in the table caption. These differences may partly reflect the two-phase fine-tuning strategy and partly the use of three classes, whereas the source study [2] used a larger number of classes. Determining how much of the improvement is attributable to the architecture rather than to the experimental setup would require re-evaluation using matched classes and data splits, which is left for future work.

Limitations and scope

The comparison presented in Table *3* provides only a partial picture, and its limitations are therefore outlined explicitly. EF-SSD and YOLO-PLNet address a different task from the present model, namely localizing lesions within an image, whereas the present system assumes that a single leaf has already been isolated. In practice, a farmer's photograph would require this localization step before classification could be applied; the proposed model is therefore more appropriately framed as a component that follows a detector such as EF-SSD, receiving its output crop and adding a confidence estimate, rather than as an alternative end-to-end pipeline.

The reported throughput requires specific qualification. The reported 137.4 frames per second was measured on Colab-allocated compute rather than on a phone or an edge device, whereas EF-SSD's 47 FPS and YOLO-PLNet's 41.3 FPS were both measured on target hardware, specifically an RTX 3080 and a Jetson Orin NX, respectively. Given the exported model's small footprint, poor on-device performance might be considered unlikely, but that remains an assumption rather than a result and is presented as such.

The PlantVillage dataset itself constitutes a limitation. Every image in the dataset was captured under uniform laboratory lighting against a clean background, and models trained exclusively on data of this kind typically perform less well when confronted with real-world visual clutter. The single external photograph tested here is encouraging but remains a single image and is therefore not a substitute for proper field validation. The relatively small size of the Healthy class (152 images compared with 1,000 images for each disease class; Table *1*) is a plausible contributor to its lower F1-score and higher uncertainty. Addressing this imbalance through more balanced data collection is therefore warranted.

The uncertainty-rejection result also requires a caveat, although of a different kind. Starting accuracy was already close to 99%, so a rejection rule that excluded the most uncertain 10% of predictions had very limited scope for improvement. The resulting change, in fact a slight decrease, reflects that ceiling rather than a weak uncertainty signal. Its primary value lies in identifying individual flagged predictions for human review, an effect that is not captured by aggregate accuracy metrics.

This work also did not include a matched ablation against a standard softmax MobileNetV3-Small trained on the same data, split, and schedule with the EDL head removed. Consequently, it is not possible to determine from this study alone whether the evidential head incurs any accuracy cost relative to a conventional classifier in this exact setup. The comparisons in Table *3* and Table *4* use models from other studies trained and evaluated on different datasets and splits, which is a weaker substitute for such an ablation. This work also did not include a head-to-head comparison with Monte Carlo Dropout [14] or deep ensembles [15], the two most common alternatives to evidential uncertainty estimation. Both were excluded by design: Monte Carlo Dropout requires dozens of stochastic forward passes per image, and deep ensembles require several independently trained networks, either of which would undermine the single-pass, real-time rationale for adopting the EDL approach. A quantitative accuracy-versus-latency comparison with these baselines is left for future work.

An additional asymmetry is that the TensorFlow Lite export used here employed dynamic-range quantization, a lighter technique than the INT8 scheme used for YOLO-PLNet's reported model size; therefore, the two model-size figures are not directly comparable.

Future work

A few things emerge as natural next steps: running the exported TensorFlow Lite model on real target hardware, such as a phone or a Jetson-class device, to obtain a throughput figure that can be compared with those reported for EF-SSD and YOLO-PLNet; testing on a field-collected image set rather than relying on PlantVillage alone; correcting the Healthy-class imbalance, whether through additional data collection or class weighting during training; and pairing this classifier with an existing detector, such as EF-SSD or YOLO-PLNet, to evaluate the performance of the combined end-to-end pipeline. The full implementation, including the data pipeline, training, evaluation, and TensorFlow Lite conversion code, is available for replication and extension [16].

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w

Conclusions

The question behind this work was whether a real-time potato leaf disease classifier could gain a calibrated sense of its own uncertainty without losing the accuracy or speed that make it worth deploying in the first place. Combining a MobileNetV3-Small backbone with an evidential deep learning head reached 98.77% test accuracy at 3.76 MB unquantized (1.06 MB as a TensorFlow Lite export) - on par with or ahead of the lightweight classifiers used for comparison - while producing a confidence estimate in the same single pass that none of those comparators offer. A simple post-hoc scaling step brought raw ECE from 0.299 down to 0.017 with no change to any prediction, and the resulting uncertainty score separated right answers from wrong ones with an AUROC of 0.727. Throughout, the aim has been to be clear about where this stops short of a deployment claim: it classifies rather than detects, its speed was measured in a development environment rather than on real hardware, and its field generalization rests on one photograph rather than a proper study. Those boundaries are what make the actual claim credible - that calibrated uncertainty and real-time, phone-scale potato disease classification are not in tension, and that a system telling a farmer both what a leaf shows and how much to trust that answer is within reach.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Khushvir Singh, Anurag Tewari

Acquisition, analysis, or interpretation of data: Khushvir Singh, Anurag Tewari

Drafting of the manuscript: Khushvir Singh, Anurag Tewari

Critical review of the manuscript for important intellectual content: Khushvir Singh, Anurag Tewari

Supervision: Anurag Tewari

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with the ICMJE uniform disclosure form, all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work.
- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The data (and/or code) supporting this study are openly available in Kaggle (https://www.kaggle.com/datasets/arjuntejaswi/plant-village) and GitHub (https://github.com/khushvir11/potatocode).

References

1. Mohanty SP, Hughes DP, Salathé M: Using deep learning for image-based plant disease detection. Frontiers in Plant Science. 2016, 7:1419. 10.3389/fpls.2016.01419
2. Zhang J, Yang X, Fu X, Wang B, Li H: LDL-MobileNetV3S: an enhanced lightweight MobileNetV3-small model for potato leaf disease diagnosis through multi-module fusion. Frontiers in Plant Science. 2025, 16:1656731. 10.3389/fpls.2025.1656731
3. Bhavani GD, Chalapathi MMV: Lightweight scalable deep learning framework for real time detection of potato leaf diseases. Scientific Reports. 2026, 16:8770. 10.1038/s41598-025-33423-7
4. Sun J, Feng Z, Han J, Xu F, Zhang H, Guo Y: YOLO-PLNet: a lightweight real-time detection model for peanut leaf diseases based on edge deployment. Frontiers in Plant Science. 2025, 16:1707501. 10.3389/fpls.2025.1707501
5. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D: D: Grad-CAM: visual explanations from deep networks via gradient-based localization. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy. 2017, 618-626. 10.1109/ICCV.2017.74
6. Sensoy M, Kaplan L, Kandemir M: Evidential deep learning to quantify classification uncertainty [PREPRINT]. arXiv. 2018, 10.48550/arXiv.1806.01768
7. Howard A, Sandler M, Chu G, et al.: Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South). 2019, 1314-1324. 10.1109/ICCV.2019.00140
8. Hughes DP, Salathe M: An open access repository of images on plant health to enable the development of mobile disease diagnostics [PREPRINT]. arXiv. 2016, 10.48550/arXiv.1511.08060
9. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L: ImageNet: a large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA. 2009, 248-255. 10.1109/CVPR.2009.5206848
10. Kingma DP, Ba J: Adam: a method for stochastic optimization [PREPRINT]. arXiv. 2017, 10.48550/arXiv.1412.6980
11. Guo C, Pleiss G, Sun Y, Weinberger KQ: On calibration of modern neural networks. Proceedings of the 34th International Conference on Machine Learning. 2017, 70:1321-1330.
12. Jacob B, Kligys S, Chen B, et al.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA. 2018, 2704-2713. 10.1109/CVPR.2018.00286
13. Hendrycks D, Gimpel K: A baseline for detecting misclassified and out-of-distribution examples in neural networks [PREPRINT]. arXiv. 2018, 10.48550/arXiv.1610.02136
14. Gal Y, Ghahramani Z: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. Proceedings of The 33rd International Conference on Machine Learning. 2016, 48:1050-1059.
15. Lakshminarayanan B, Pritzel A, Blundell C: Simple and scalable predictive uncertainty estimation using deep ensembles. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates, Inc., Red Hook, NY; 2017. 6405-6416. 10.5555/3295222.3295387
16. Singh K: Potato leaf disease EDL classifier—code repository. GitHub. (2026). Accessed: September 29, 2026: https://github.com/khushvir11/potatocode.

How to cite this article:
Singh K, Tewari A (September 29, 2026) Real-Time, Uncertainty-Aware Classification of Potato Leaf Diseases Using Evidential Deep Learning: A Lightweight MobileNetV3-Small Framework for Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00301-w. DOI https://doi.org/10.7759/s44389-026-00301-w