RIT

Reproducing DeepWeeds: Weed Classification with Transfer Learning and Grad-CAM Interpretability

Srujan Vaddiparthi

Data Science Co-op @ ITT Goulds Pumps

MS Data Science '26 | BE Mech Eng.+Robotics '23

Applications of ML for Remote Sensing

IMGS 589

Agenda

1. Motivation & Problem Statement
2. Dataset & Preprocessing
3. Tech Stack & Hardware
4. Modeling Approach
    a. Transfer Learning with ResNet-50
    b. Training strategy & augmentations
5. Results
    a. Accuracy, Confusion Matrix
    b. Class-wise performance
6. Grad-CAM Interpretability
    a. Correct predictions
    b. Misclassified examples
7. Discussions & Limitations
8. Future Work
9. Conclusion

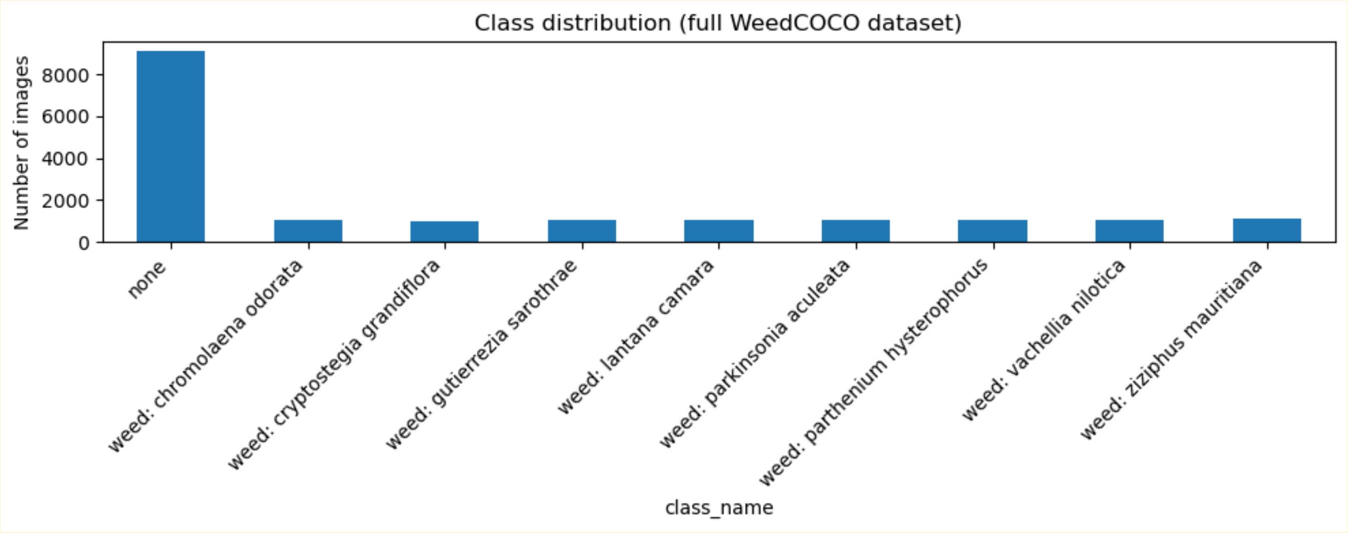
Motivation: Why Weed Detection?

- Invasive weeds cause ecological and agricultural damage.
- DeepWeeds(Olsen, S. R. (2019)) dataset is a strong benchmark for weed-classification.
- Need models that are **accurate** *and* **interpretable**.
- A reliable AI system for weed identification:
  - Reduce water waste,
  - Reduce unnecessary chemical use,
  - Make lawn maintenance more affordable and sustainable.
- Interpretability matters for:
  - Safety (especially with robotics),
  - Environmental decision-making,
  - Trust and transparency in ecological applications
- How does this fit into a remote sensing course?
  Techniques here (classification + interpretability) directly transfer to UAV and drone-based vegetation mapping, crop monitoring, and ecological surveying.

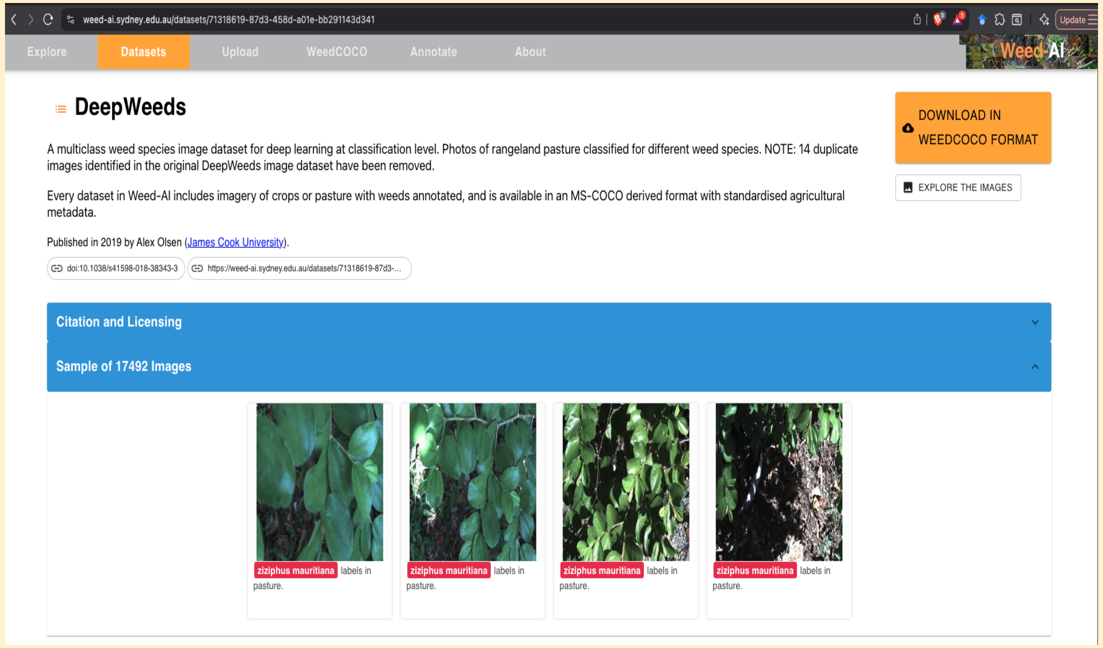
Lantana Camara

Dataset

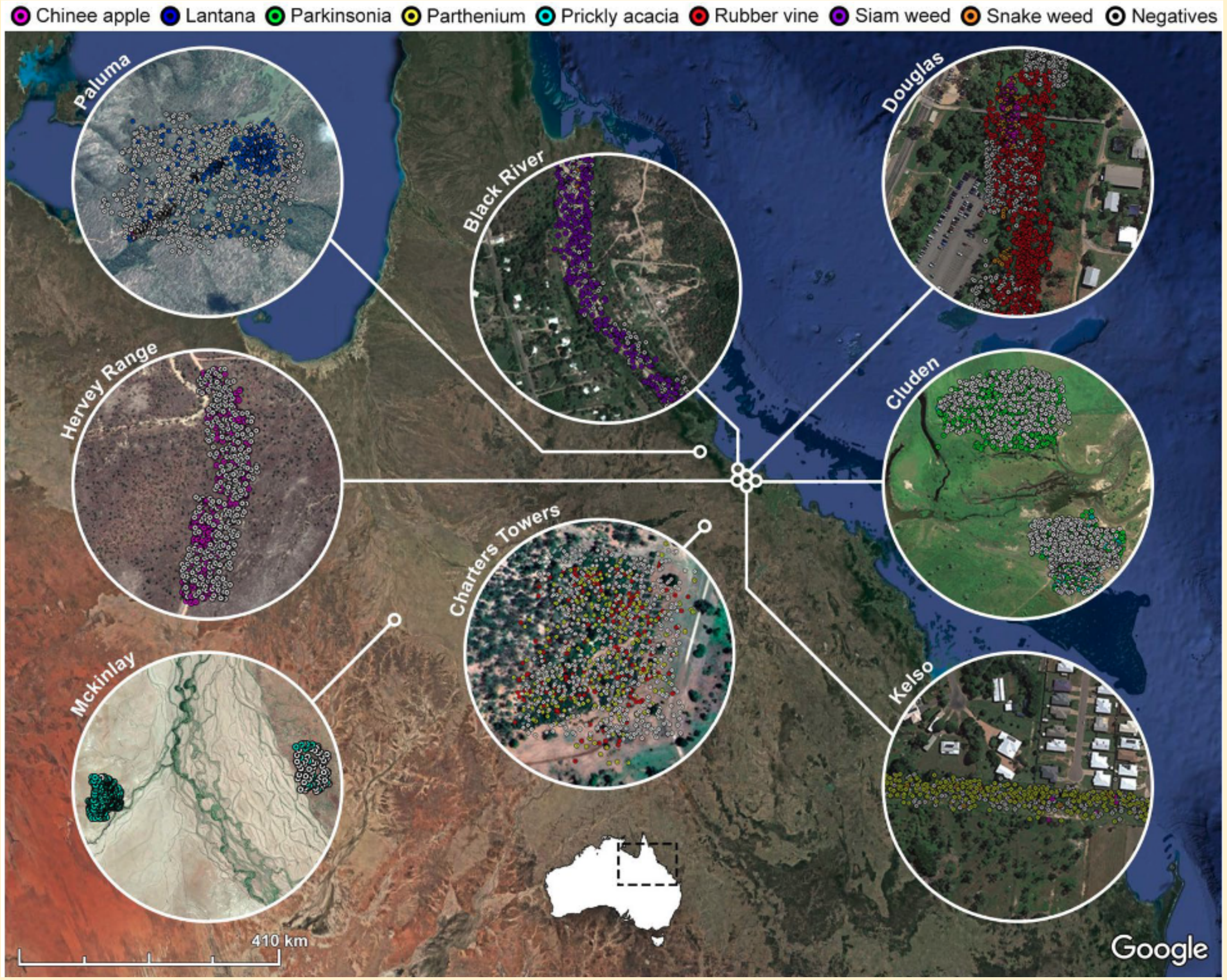
- Source: Weed-AI repository
- 17,492 RGB images. [image folder].
- 8 weed species + “none” class. [weedcoco.json annotation file (class labels + metadata)].
- Intra-class variation: lighting changes, growth stages, cluttered vegetation backgrounds.



Dataset



Weed-AI website: DeepWeeds dataset in MSCOCO format.

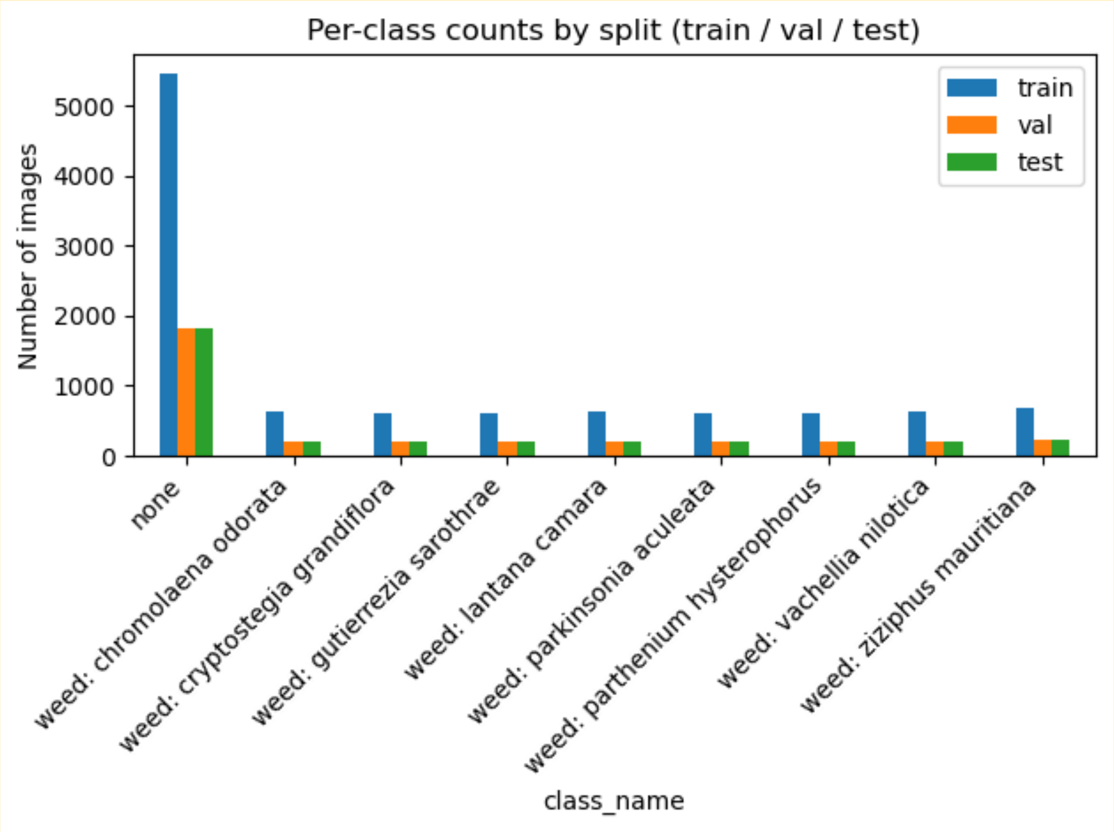


Geographical Distribution of weeds across northern Australia
Image from DeepWeeds paper (Olsen 2019).

Preprocessing

```
DataFrame shape (rows = images with labels): (17492, 4)
```

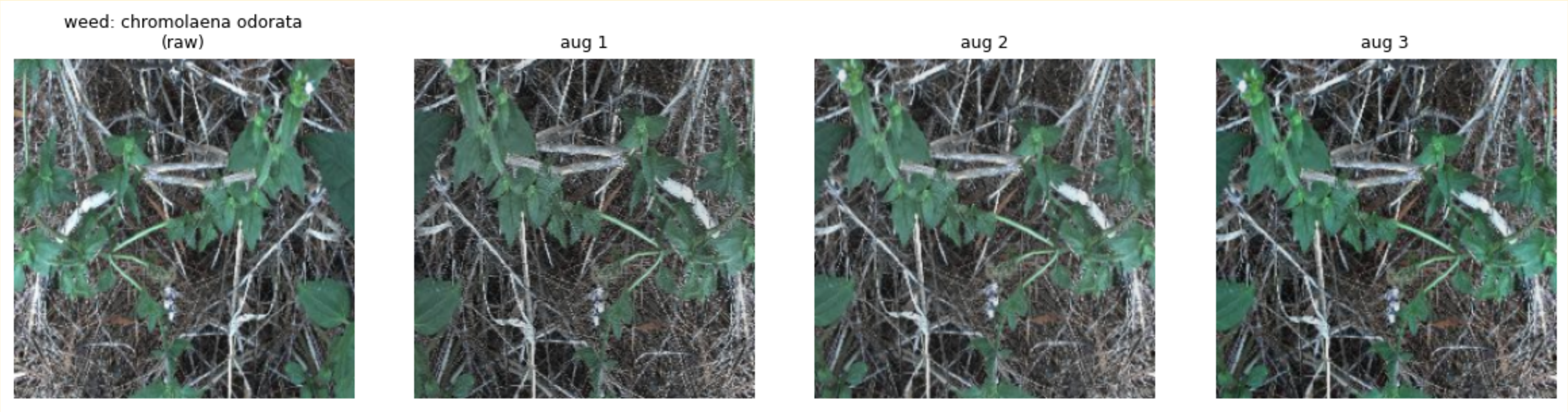
	image_id	file_name	category_id	class_name
0	0	images/44179195d7afe4ce5cb9.jpg	0	weed: ziziphus mauritiana
1	1	images/d9c8a6da4da6af0f22ff.jpg	0	weed: ziziphus mauritiana
2	2	images/787ee4d4bd5854e700d5.jpg	0	weed: ziziphus mauritiana
3	3	images/8fafed4b9942a3b0be26.jpg	0	weed: ziziphus mauritiana
4	4	images/26ff7c309cbdfd00017f.jpg	0	weed: ziziphus mauritiana



- Converted WeedCOCO JSON → DataFrame.
- Performed a stratified split (60/20/20) to preserve class balance.
- Standardized using ImageNet normalization (μ, σ) for Transfer Learning.
  IMAGENET_MEAN = [0.485, 0.456, 0.406]
  IMAGENET_STD = [0.229, 0.224, 0.225]

Preprocessing

- Applied targeted augmentations for training:
  - Resize + RandomResizedCrop (224×224) [aug1]
  - Horizontal Flip [aug2]
  - Color jitter [aug3]
- No heavy augmentation on Validation & Test
  Only Resize + Normalization.



Tech Stack & Hardware

Tech Stack

- Python 3.11
- PyTorch for training & inference
- Torchvision pretrained ResNet-50
- NumPy / Pandas for data structuring
- Matplotlib for visualization
- pytorch-grad-cam for interpretability
- scikit-learn for metrics & confusion matrix

Hardware

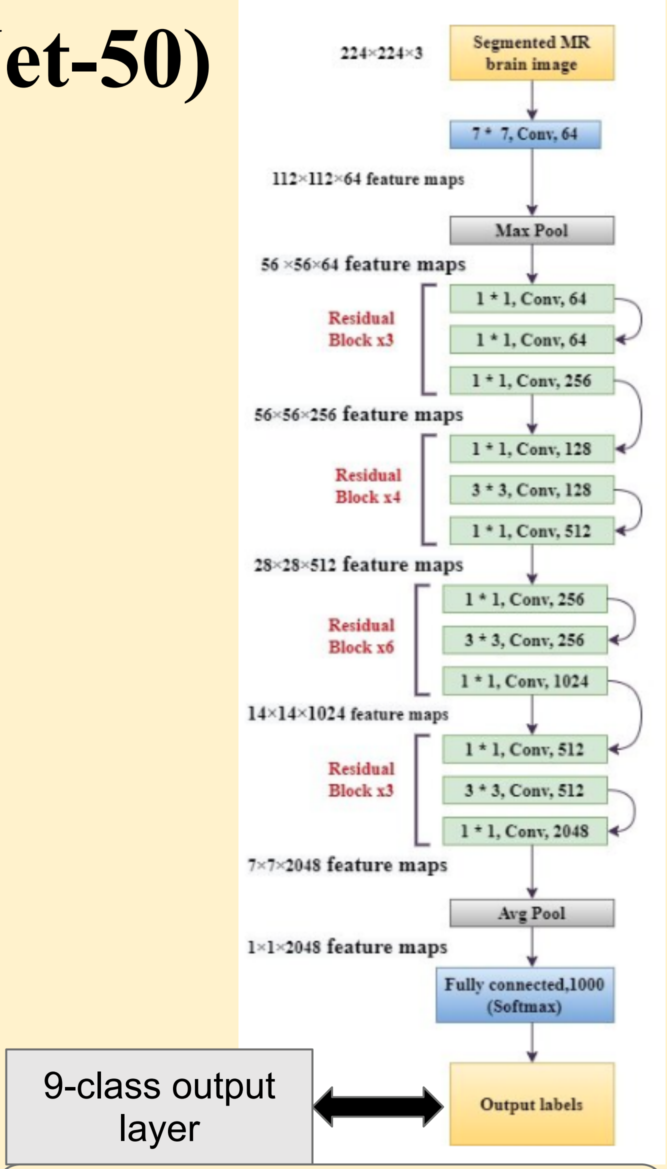
- Apple M1/M2 CPU + MPS acceleration (Mac)
- Training time: ~40 minutes for 11 epochs
- Batch size: 32
- Image size: 224 × 224

Project Structure

- Simple WeedCOCO → DataFrame → DataLoader pipeline
- Reproducible training loop
- Saved model checkpoints & Grad-CAM outputs

Modeling Approach (Transfer Learning with ResNet-50)

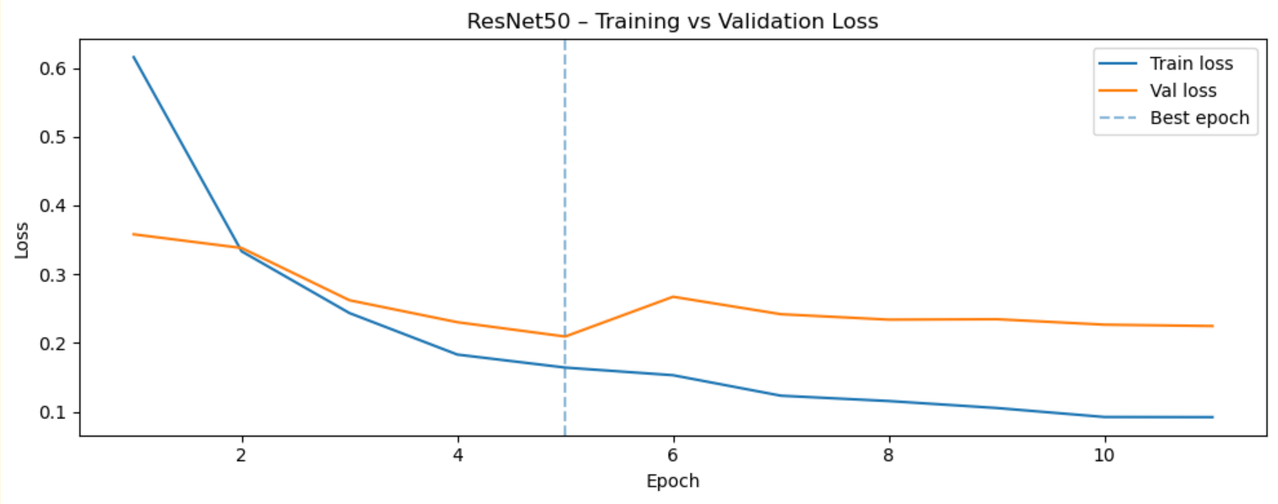
- **Base architecture**: ResNet-50 (pretrained on ImageNet).
- **Reason for choosing**:
  - strong baseline shown in DeepWeeds paper
  - ImageNet weights provide robust low-level feature extractors
- **Modification**:
  - Replaced the final fully connected layer with a 9-output layer.
- **Why transfer learning?**
  - Training from scratch (as in DeepWeeds) requires 50-100 epochs + large compute.
  - Using pretrained features allows faster convergence (10-12 epochs).
  - Reduces overfitting risk on moderate-sized datasets.
- **What feature extractors already "know"**:
  - Edges, shape, corners
  - Textures, repeating patterns
  - Color gradients
- **What the model learns from weed data**:
  - Leaf morphology, branching patterns, texture, species-specific shape cues.

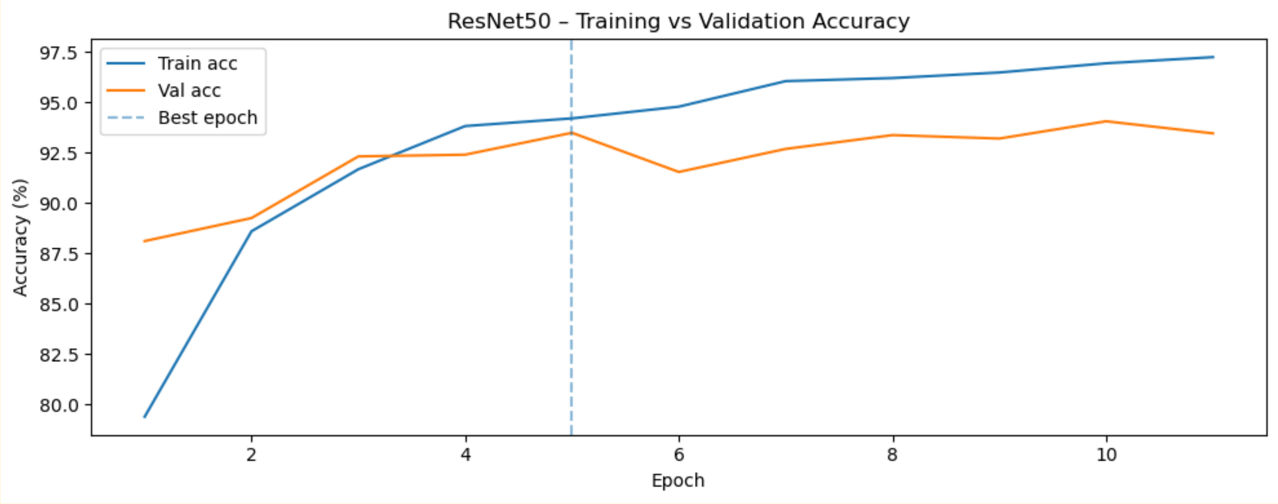


models.resnet50(pretrained=True)

Modeling Approach (Training Strategy)

- **Optimizer:** Adam (learning rate α = 1e-4)
- **Loss Function**: CrossEntropyLoss (log loss)
- **Training Setup:**
  - Full fine-tuning (all layers updated)
  - 60/20/20 stratified split.
- **Epochs**:
  - Max 30 epochs
  - Early stopping (patience = 5)
  - Best model selected based on validation loss
- **Why early stopping?:**
  - Prevents overfitting
  - Reduces training time
- **Observed behavior:**
  - Stable convergence
  - Train accuracy rose from ~79% → ~ 97%
  - Validation accuracy peaked at ~94%
- **Robustness check** with a different random-seeded split.





Modeling Approach (Training Strategy by DeepWeeds)

Reference only

- **CNN:** Resnet-50 from scratch
- **Pretrained weights**: NO
- **Epochs:** 100
- **Early stopping:** NO
- **Optimizer:** Stochastic Gradient Descent (SGD) with momentum (not Adam)
- **Learning rate schedule:** manually decayed over time.
- **Batch size**: 32 (same)
- **Augmentations:**
  - Resize 256 → 224
  - Horizontal flip
  - Light brightness jitter
- **Split: 8 geographical sites**
  - 7 sites for training
  - 1 held-out site for validation
  - Rotated in a leave-one-site-out (LOSO) approach
- **Hardware:** NVIDIA P100 GPU cluster (much more compute)

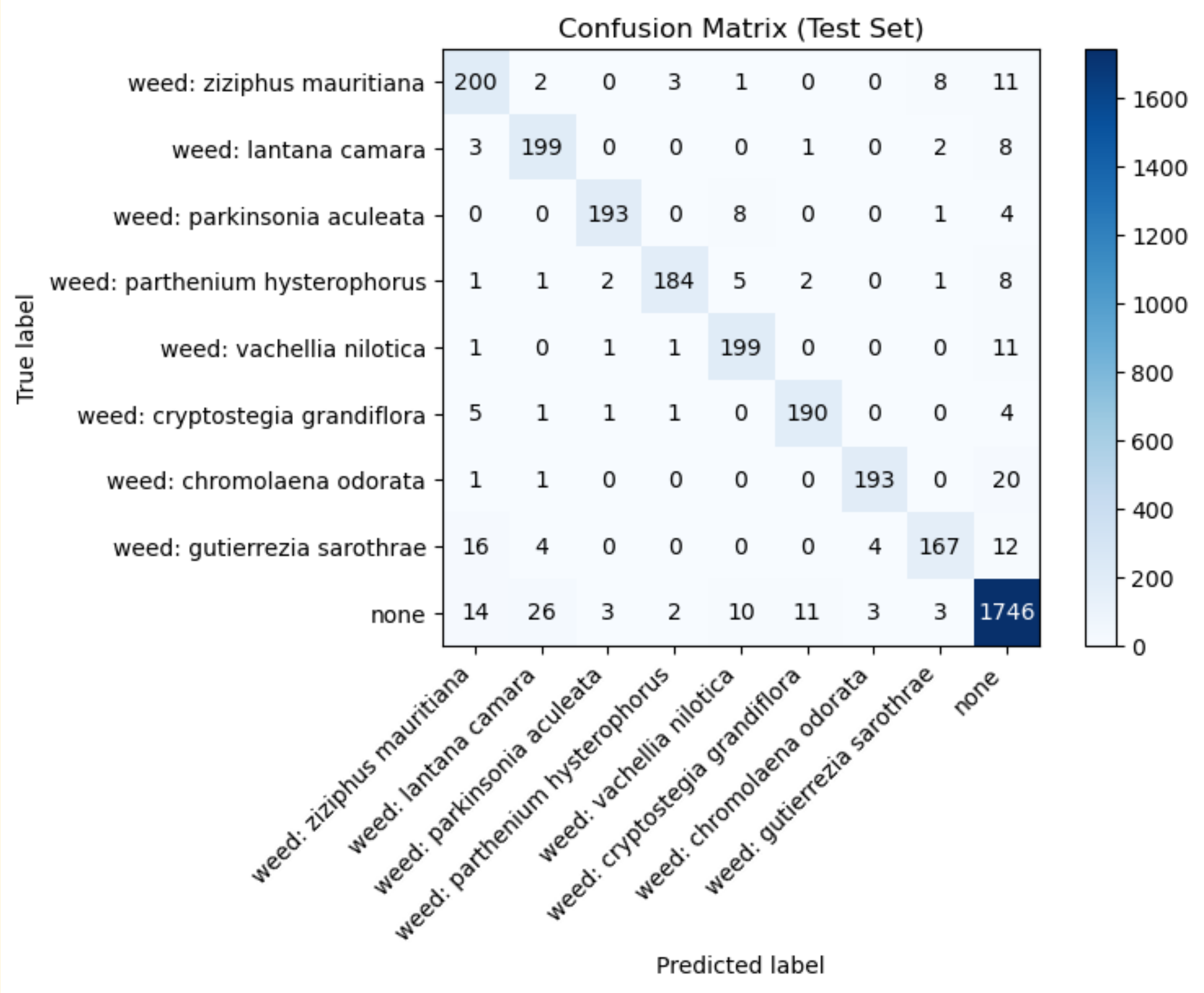
DeepWeeds used a resource-heavy training setup: 100 epoch scratch training with no transfer learning and a geographically stratified evaluation. For this course, a practical and modern transfer-learning pipeline was implemented. Using ImageNet weights and early stopping, similar accuracy was achieved in only 11 epochs and with dramatically less computation.

Results – Accuracy & Evaluation

- **Test Accuracy:** 93.48% | 95.7% deepweeds
- **Repeated split accuracy**: 93.60%
- **Macro F1-score**: ~0.92
- **Weighted F1-score**: ~0.93
- Performance consistent across two random splits (robustness check)
- Most misclassifications stem from visually similar species or heavy background clutter.

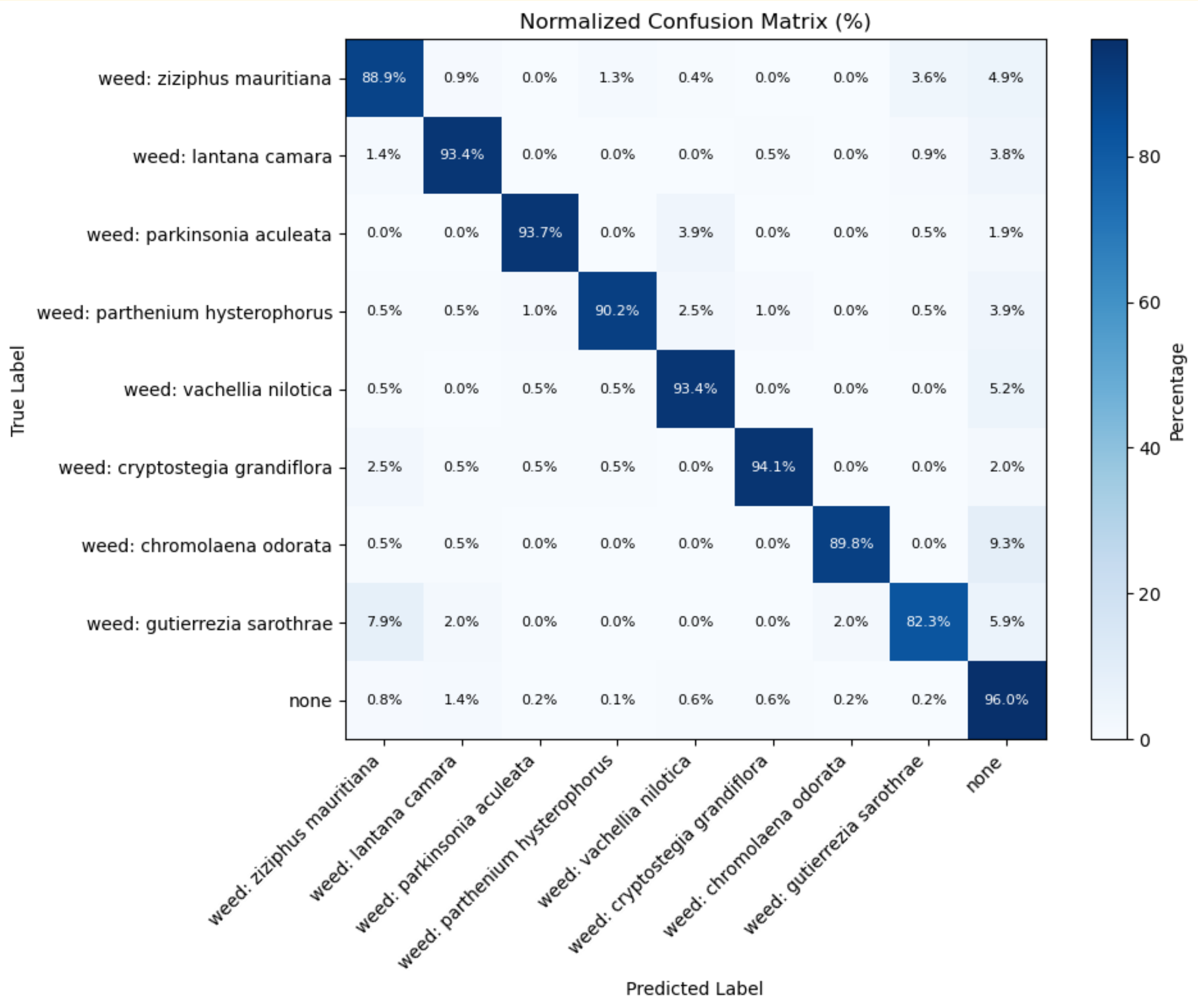
Classification report:

	precision	recall	f1-score	support
weed: ziziphus mauritiana	0.83	0.89	0.86	225
weed: lantana camara	0.85	0.93	0.89	213
weed: parkinsonia aculeata	0.96	0.94	0.95	206
weed: parthenium hysterophorus	0.96	0.90	0.93	204
weed: vachellia nilotica	0.89	0.93	0.91	213
weed: cryptostegia grandiflora	0.93	0.94	0.94	202
weed: chromolaena odorata	0.96	0.90	0.93	215
weed: gutierrezia sarothrae	0.92	0.82	0.87	203
none	0.96	0.96	0.96	1818
accuracy			0.93	3499
macro avg	0.92	0.91	0.92	3499
weighted avg	0.94	0.93	0.93	3499



Results – Confusion Matrix Comparison

Reference only



	Chinee apple	*Lantana*	*Parkinsonia*	*Parthenium*	*Prickly acacia*	*Rubber vine*	*Siam weed*	*Snake weed*	*Negatives*
Chinee apple	**88.5**	1.78	0.00	0.44	0.18	0.18	0.27	3.37	5.33
Lantana	0.56	**95.0**	0.00	0.00	0.00	0.09	0.28	0.94	3.10
Parkinsonia	0.10	0.00	**97.2**	0.10	1.26	0.00	0.00	0.00	1.36
Parthenium	0.10	0.20	0.10	**95.8**	0.88	0.10	0.00	0.29	2.54
Prickly acacia	0.00	0.00	0.56	0.66	**95.5**	0.00	0.00	0.09	3.20
Rubber vine	0.79	0.50	0.10	0.10	0.00	**92.5**	0.20	0.40	5.45
Siam weed	0.00	0.19	0.00	0.00	0.00	0.00	**96.5**	0.09	3.26
Snake weed	4.13	1.77	0.00	0.30	0.20	0.10	0.30	**88.8**	4.43
Negatives	0.46	0.48	0.14	0.20	0.55	0.03	0.21	0.37	**97.6**

Table 3. The confusion matrix (%) achieved by the ResNet-50 model on the test subsets for the five cross validated folds.

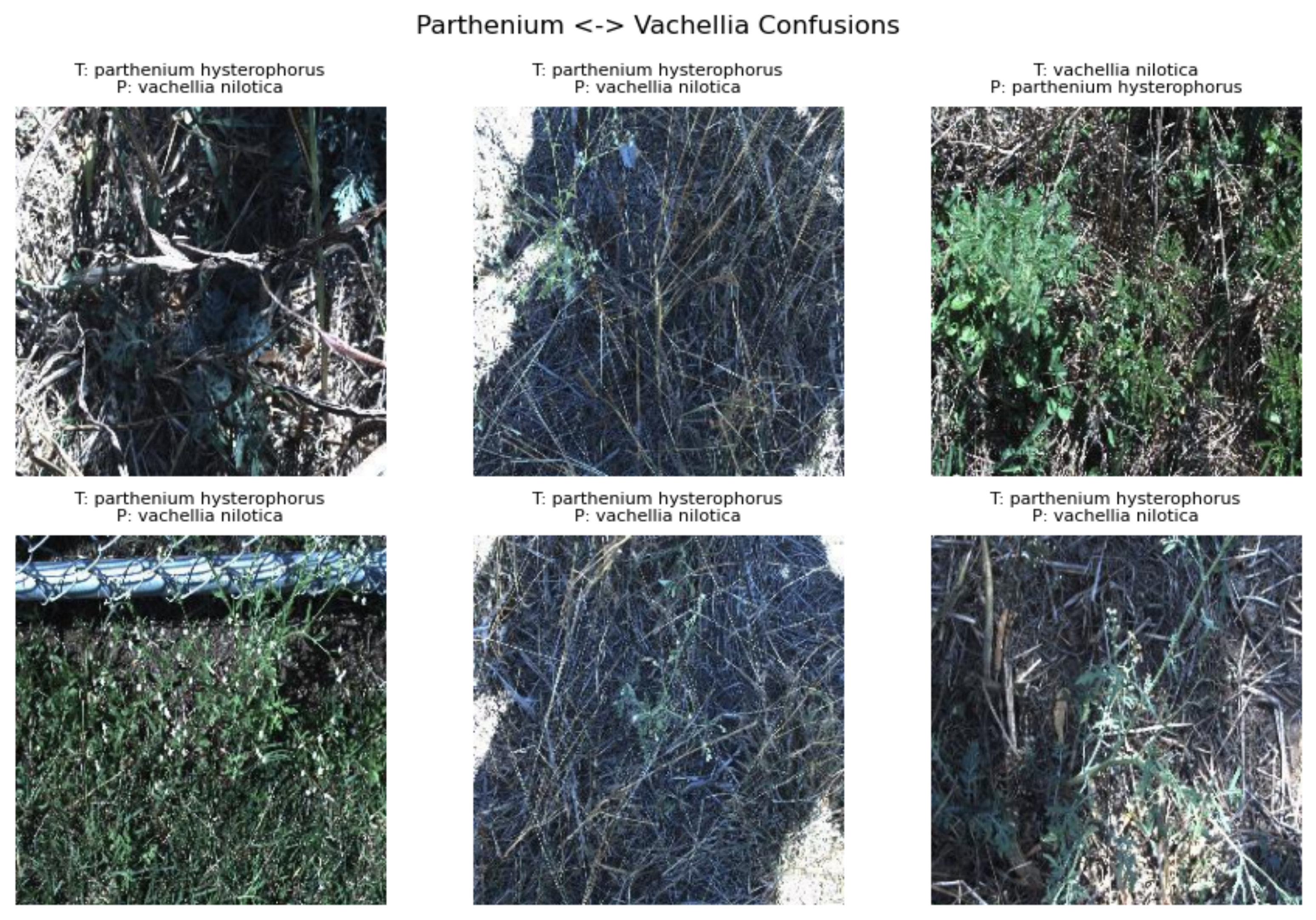
Results – Class-wise performance | Misclassification pattern

- **Most mistakes happen between "weed" and "none"**, especially when the weed is small, occluded, or blends into background vegetation.
- **Fine-grained confusions occur between visually similar species**, which is expected in cluttered natural scenes.
- **Distinctive species show very few errors**, meaning the model relies heavily on clear morphological cues.
- **Overall error patterns are intuitive** and match the limitations of real-world field imagery (lighting, clutter, partial visibility).

Results – Class-wise performance | Misclassification pattern

Reference only

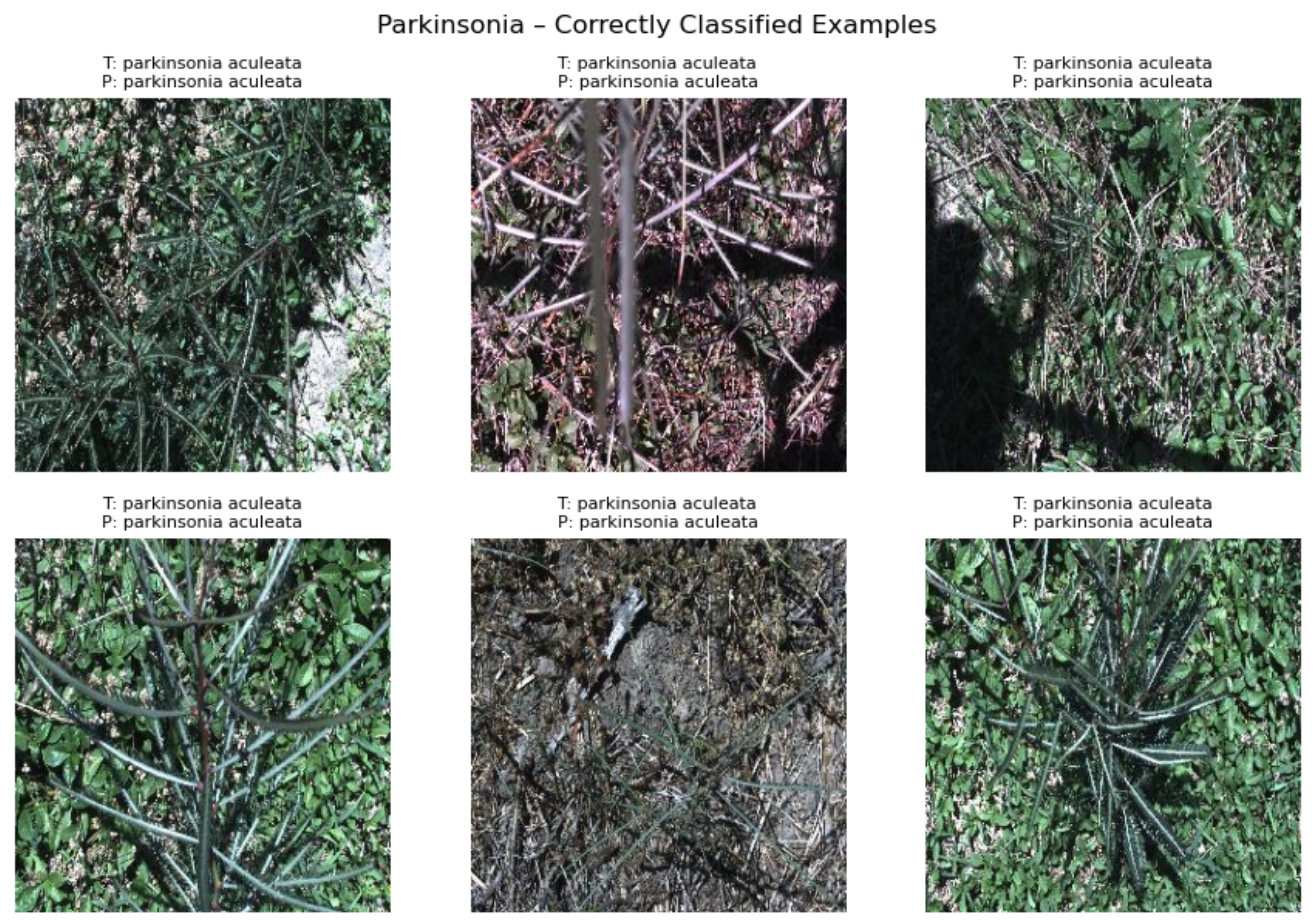
Confusion between similar looking weeds:



Results – Class-wise performance | Misclassification pattern

Reference only

Distinctive species – Correct Classification



Results – Class-wise performance | Misclassification pattern

Reference only

Misclassification of Weed → None class

Results – Accuracy & Evaluation
Metric definitions
Reference only

Overall accuracy:

Measures the proportion of correctly classified test samples:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ test\ samples}$$

This metric provides a coarse indication of performance but does not account for class imbalance, and therefore must be supplemented by class-specific measures.

Precision, Recall, and F1-score (Per Class):

For each of the nine classes, the following standard classification metrics are reported:

- Precision: the proportion of samples predicted as class i that truly belong to class i
- Recall: the proportion of true class i samples that are correctly identified.
- F1-score: the harmonic mean of precision and recall.

These metrics provide insight into class-specific behavior, particularly for species that may be visually similar or frequently confused.

Macro-Averaged Metrics:

Macro-averaged precision, recall, and F1-score compute the unweighted mean across classes:

$$F1_{macro} = \frac{1}{C}\sum_{i=1}^{C} F1_i \text{ where C=9 classes.}$$

Macro-averaging treats each class equally, providing an evaluation that is not dominated by the majority class.

Weighted-Averaged Metrics:

Weighted metrics incorporate class frequency:

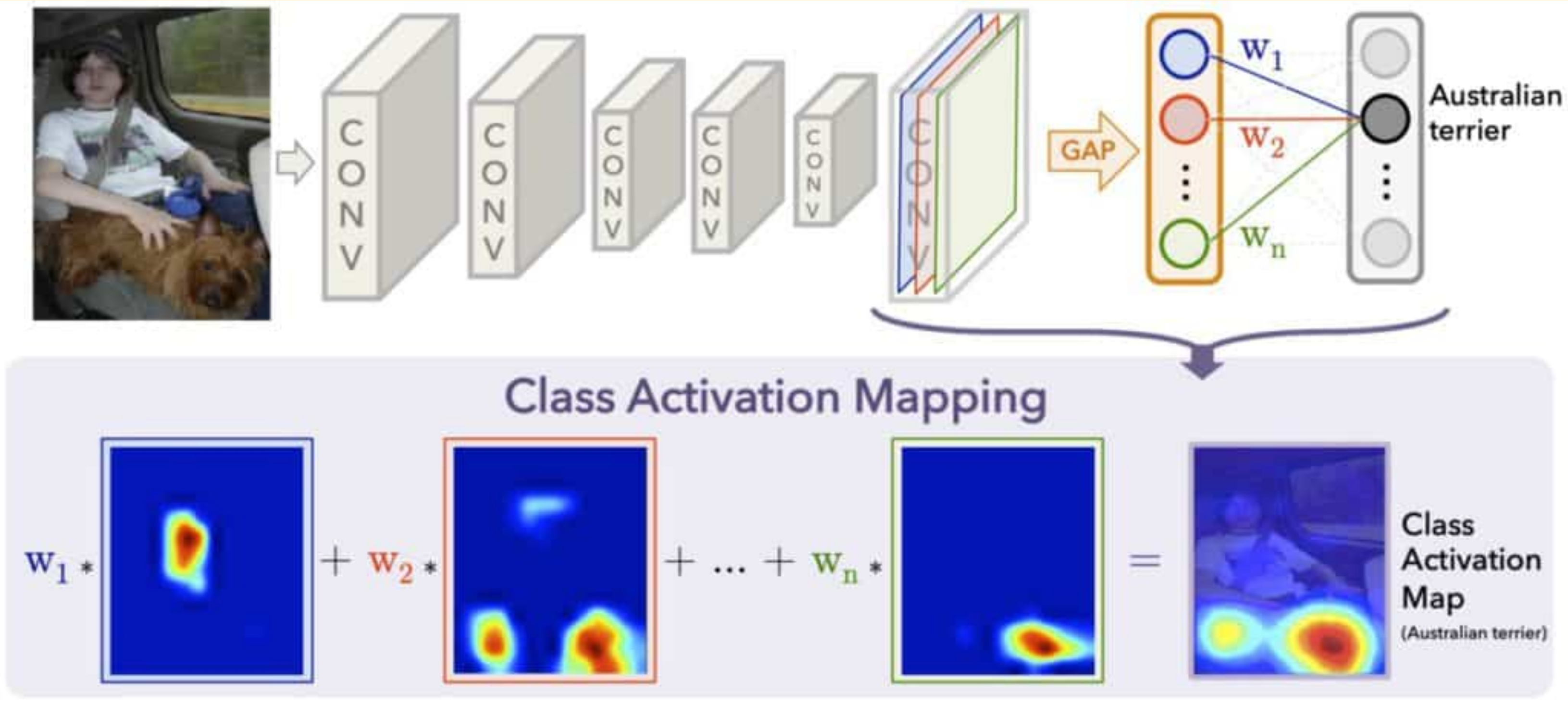
$$F1_{weighted} = \sum_{i=1}^{C} w_i \cdot F1_i \text{ where } w_i = \frac{support_i}{N}.$$

Weighted scores reflect real dataset proportions and provide additional insight when imbalance is present.

Grad-CAM Interpretability

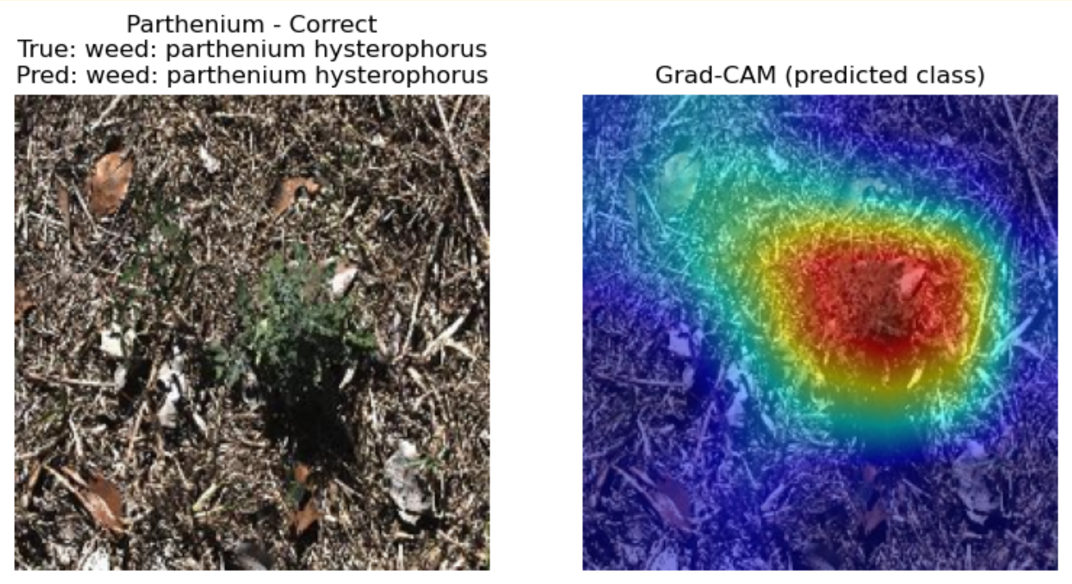
Gradient-weighted Class Activation Map (Grad-CAM)

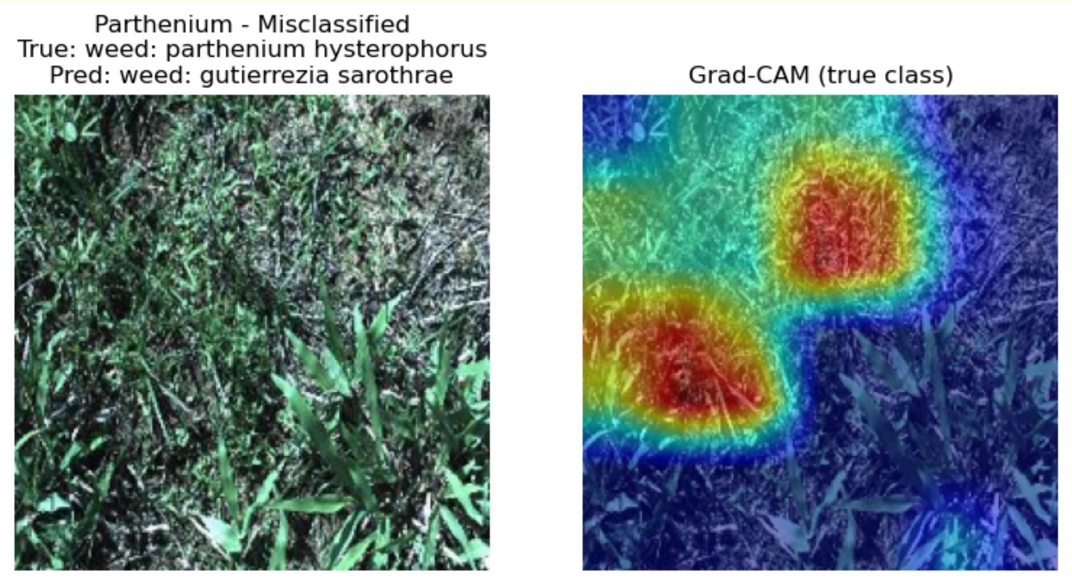
- Visual explanation of *where* the CNN focuses during prediction.
- Uses gradients from the final conv layer
- Helps verify if the model attends to leaf structure or irrelevant background
- Essential for biological/ecological trust and debugging misclassifications.

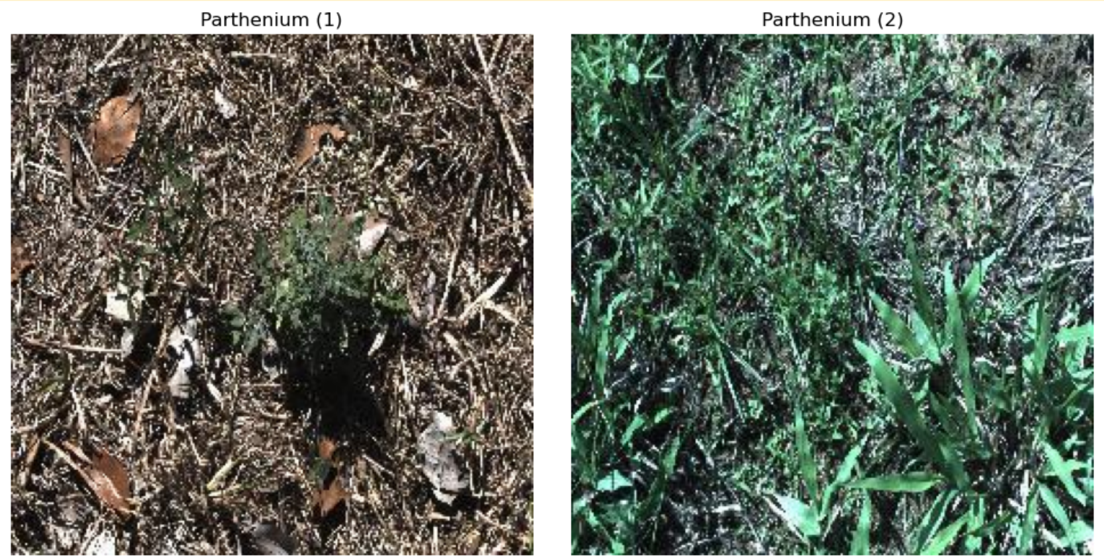


Grad-CAM creates a heatmap by combining feature activations with their importance for the predicted class, showing which image regions most influenced the model's decision.

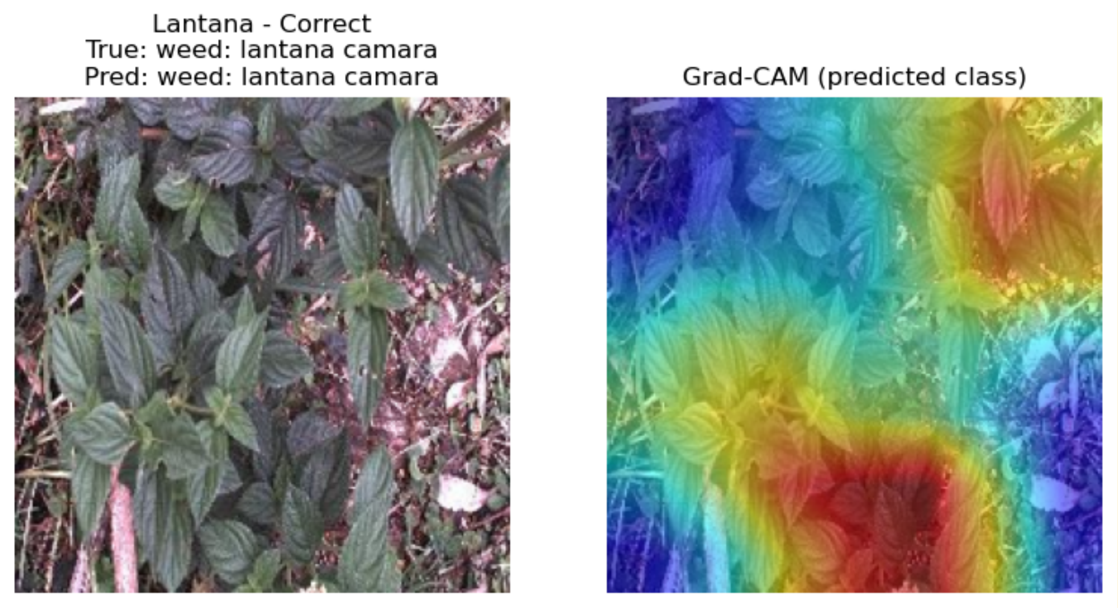
Grad-CAM Interpretability







Grad-CAM Interpretability



Discussions

- The ResNet-50 model achieved strong and stable performance (~93–94%), consistent with DeepWeeds' original findings.
- Grad-CAM showed that the network generally focuses on leaf structure, texture, and branch clusters, confirming meaningful feature learning.
- Misclassifications mostly happened under cluttered vegetation, poor lighting, or between visually similar species.

Limitations

- **RGB-only dataset**
  - Limited spectral information → harder to distinguish species with similar color/texture.
- **No spatial annotations:**
  - No bounding boxes, or segmentations masks → cannot directly evaluate crown localization.
- **One primary train/val/test split:**
  - A second seed confirmed stability, but full K-fold or multi-site splits were not implemented.
- **Transfer learning vs. “from scratch” mismatch:**
  - DeepWeeds trained from scratch; this project used ImageNet features → not a perfect comparison. Heuristic comparison only.
- **Interpretability constraints:**
  - Grad-CAM sometimes highlighted background regions in misclassified images → attention may drift under clutter.
- **Lack of time ;-;**

What this means:
The pipeline is strong, replicable, and interpretable,
but not yet crown-aware, not yet spatial, and not yet field-robust → perfect setup for **future work**.

Future Work

1. **Crown-Center Localization:**
    - Use Grad-CAM (or other methods) + simple masks to estimate where the weed meets the soil.
    - A small step towards more useful outputs for field or backyard applications.
2. **Basic Segmentation Experiments:**
    - Lightweight models (U-Net/SAM masks) on a small annotated subset.
    - Help model focus on actual weed region, not the background.
3. **Robustness Checks:**
    - Test how the model behaves under changes in lighting, blur, or partial occlusion.
    - Important for real-world images that are less “clean” than the dataset.
4. **Edge Deployment:**
    - Explore running the model in compressed form (FP16/ONNX/pruning).
    - Useful for low-power devices like Raspberry Pi or small rover/drones.
5. **One Additional Architecture:**
    - Compare ResNet-50 with one modern model (eg. Swin-Tiny or EfficientNet-B0).
    - A simple extension that adds scientific value without large compute needs.
6. **Broader Curiosity-Driven Extensions:**
    - These ideas are exploratory, aimed at learning segmentation and tackling harder perception problems — not only for weeds but any object-centric tasks in robotics, ecology, or remote sensing where locating the “important region” matters.

Conclusion

Project Summary:

- Reproduced the DeepWeeds classification pipeline using a modern PyTorch workflow.
- Fine-tuned ResNet-50 with ImageNet weights → achieved a strong performance (~93% accuracy).
- Built a clean Weed-AI → DataFrame → DataLoader pipeline.

Key Findings:

- Reliable classification of visually similar weed species by the model.
- Misclassifications follow intuitive patterns (background clutter, overlapping foliage).
- Grad-CAM showed that the model focuses on meaningful plant structures, not noise.

Takeaway:

This work provides a solid, interpretable baseline for real-world weed identification, and establishes a foundation for extensions such as crown localization, segmentation, and lightweight deployment.

Closing thoughts:

The broader value of this project lies not just in weeds, but in learning how visual models see, a skill applicable to robotics, ecology, remote sensing, and many future problems.

Thank you!
Questions?...