IFN680: AIML

Brisbane Flora and Fauna Society

FloraSight

AI-Powered Plant Classification

TABLE OF CONTENT

MESSAGE FROM OUR ANALYST

Dear Members of the Brisbane Flora and Fauna Society,

I am thrilled to present this report on the FloraSight Classifier project, a groundbreaking initiative that aims to revolutionize the way we monitor and protect Brisbane's unique biodiversity. As someone deeply committed to leveraging technology for environmental conservation, it has been a privilege to lead the analytical aspects of this project.

The project involved the development of two machine learning models: a 'Teacher Model' and a 'Student Model.' The Teacher Model was trained on a carefully curated dataset and serves as our gold standard for plant classification. The Student Model, on the other hand, was fine-tuned using a combination of labeled and pseudo-labeled data, the latter being generated by the Teacher Model. This semi-supervised approach not only improves the model's accuracy but also makes it adaptable to new, unlabeled data in the future.

I am pleased to report that our initial tests have been highly promising. The FloraSight Classifier has demonstrated a high level of accuracy in identifying both healthy plants and weeds, thereby making it a valuable tool for your conservation efforts. This technology has the potential to engage community participation at an unprecedented scale. Imagine a future where any volunteer can snap a photo of a plant, receive immediate classification from our algorithm, and contribute to a real-time monitoring system.

I would like to extend my gratitude to all the members of BFFS for your unwavering support and collaboration. Your expertise and insights have been invaluable in shaping this project. Together, we are taking a significant step towards a more sustainable and biodiverse Brisbane.
Thank you for entrusting me with this important initiative. I look forward to continuing our work together to bring the FloraSight Classifier to full fruition.

Sincerely,
Naman Khosla
Lead Analyst, FloraSight Classifier Project

INTRODUCTION

Understanding the multifaceted types of flora, such as flowers and weeds, is pivotal across various sectors like agriculture, biology, and consumer horticulture. The Brisbane Flora and Fauna Society (BFFS) has long stood as a vanguard in the conservation and study of Brisbane's rich biodiversity, striking a balance between urban development and ecological sustainability. However, one of the pressing challenges faced by BFFS is the scalable and accurate identification of native plant species as well as invasive weeds within the parklands of Brisbane. Traditionally, this task has been executed by trained experts who label, monitor, and remove specific plant species. While effective, this manual approach is not scalable and comes with its own set of challenges, primarily the extensive time and financial investment in training experts.

To address these limitations, BFFS has envisioned an innovative solution: the implementation of Artificial Intelligence (AI) to automatically classify plant species based on photographs. This study aims to explore the feasibility of this vision by designing and evaluating a machine learning model for the task.

The deliverables for this project include:

- A machine learning model trained to identify 5 different healthy plant species and 5 different weed species.
- An evaluation of the model's performance based on a held-out dataset, which will be kept undisclosed for unbiased assessment of the model.
- Recommendations for the practical deployment of such a model and any limitations observed during the study.

For this investigation, a PyTorch implementation of the ResNet18 architecture has been chosen, leveraging its pre-trained weights from ImageNet to facilitate the training process. The model will undergo adjustments to its final layer to accommodate the classification task involving 10 different plant species.

This proposal elaborates on the methodology, constraints, findings, and recommendations stemming from this study, with the goal of providing BFFS with a technology-driven, scalable solution for plant species identification.

STUDENT-TEACHER MODEL

In machine learning, semi-supervised learning techniques utilize both labeled and unlabeled data to train models. This is particularly advantageous when labeled data is limited, as is the case with the plant species dataset provided by BFFS. One effective semi-supervised learning strategy is the Student-Teacher model architecture, which we have selected for this project.

What is the Student-Teacher Model?

In this framework, two neural networks work in tandem: the "Teacher" and the "Student". Initially, the Teacher model is trained on the available labeled data to achieve a robust performance level. This trained Teacher then infers pseudo-labels for the unlabeled dataset. The Student model is subsequently trained on this newly annotated dataset, inheriting the Teacher's knowledge while also learning from the ambiguities and complexities present in the unlabeled data.

Why Select the Student-Teacher Model?

1. **Maximization of Available Data**: Given that BFFS has a substantial amount of unlabeled data, this architecture allows us to leverage that to improve the Student model's performance and robustness.
2. **Cost-Effectiveness**: Using pseudo-labels eliminates the need for human experts to label all the available data manually, thereby reducing costs and time.
3. **Improved Generalization**: Training the Student model with both labeled and pseudo-labeled data tends to make the model more resilient to overfitting, thus potentially improving its generalization to new, unseen data.
4. **Flexibility**: The Student-Teacher framework provides the flexibility to continuously update and improve the model. As more labeled data becomes available, the Teacher model can be updated, and new pseudo-labels can be generated for further training.
5. **Alignment with Project Constraints**: The Student-Teacher model can be easily implemented within the constraints specified by BFFS, including the use of a ResNet18 architecture in PyTorch with pre-trained weights from ImageNet.

In summary, the Student-Teacher model was chosen to exploit the full potential of both the labeled and unlabeled datasets provided, offering a scalable and efficient way to approach the plant species classification problem faced by BFFS.

ADDRESSING TASK 1: PERFORMANCE WITH ONLY LABELLED DATA

Statistics about the Training and Validation Dataset

We initiated our project by preparing a dataset comprising 10 different plant species: 5 healthy species and 5 weed species.

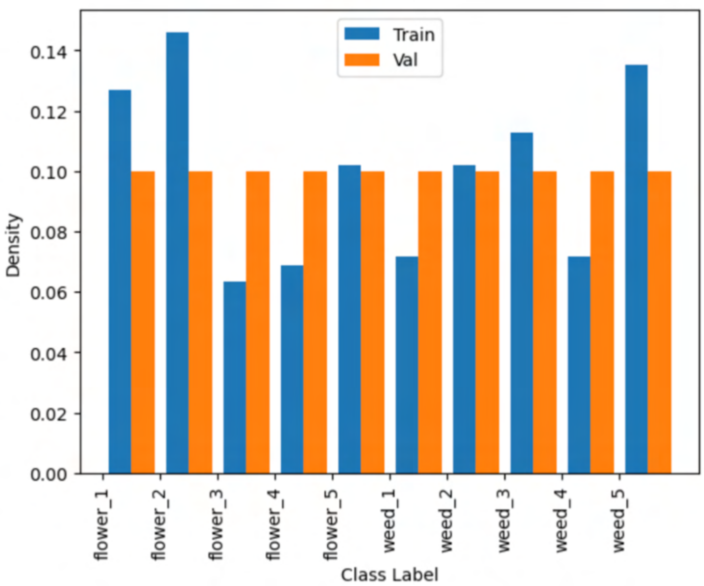
The dataset includes:

Labelled Data: 548 images with expert-verified labels, distributed across the 10 plant species.
Unlabelled Data: 426 images of plant species, awaiting classification.

Data Splitting
We split our dataset into training and validation sets. The training set comprises 63% of the data, ensuring that each class has 20 images for validation. This ensures a balanced and representative training set.

- **Total Number of Images in the Training Dataset**: 348
- **Total Number of Images in the Validation Dataset**: 200

Data Exploration
Upon examining the dataset, we discovered the following key details:
Number of Classes: There are a total of 10 plant species in the dataset.
Class Labels: The classes include 'flower_1', 'flower_2', 'flower_3', 'flower_4', 'flower_5', 'weed_1', 'weed_2', 'weed_3', 'weed_4', and 'weed_5'.



TRAINING THE TEACHER MODEL

The Teacher Model serves as the foundational supervised learning component in our semi-supervised learning architecture. Designed to be a highly efficient and versatile image classifier, the Teacher Model was implemented using a fine-tuned ResNet-18 architecture.

Highlights and Implementation Details

To train the machine learning model for this project, a thoughtful selection of hyperparameters and data augmentations was made to optimize performance. Below are the details:

- **Learning Rate:** 0.0007
- **Momentum:** 0.9
- **Batch Size:** 8
- **Number of Epochs:** 12
- **Weight Decay:** 0.001
- **Data Augmentations:**
  - **RandomHorizontalFlip:** To introduce variability and robustness.
  - **RandomVerticalFlip:** To further add to the data's complexity.
  - **RandomRotation(90):** To make the model invariant to orientation changes.

- **Optimizer**: We used a standard optimization technique with momentum and weight decay to update the weights of the neural network.

- **Training and Validation Phases**: For each epoch, we cycled through training and validation phases, calculating losses and accruing accuracy metrics for both phases.

- **Training Phase**: The model parameters were updated through backpropagation.

- **Validation Phase**: Model evaluation was conducted without updating parameters, serving as an unbiased performance gauge.

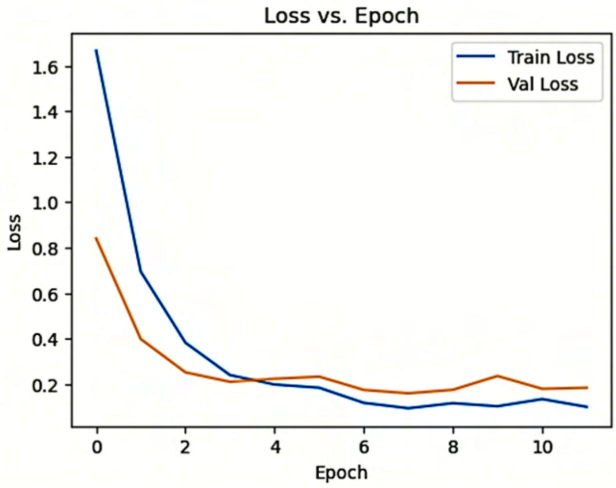
- **Performance Logging**: We employed Weights and Biases (wandb) for efficient logging and tracking of model performance across epochs. This helped in real-time monitoring of training and validation losses and accuracies.

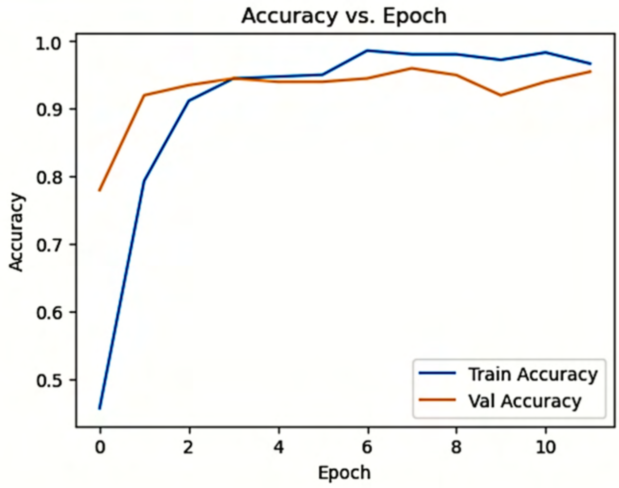
- **Loss and Accuracy Tracking**: Lists were maintained to keep a record of the training and validation loss and accuracy values for each epoch, aiding in post-analysis and model evaluation.

By maintaining a judicious balance of parameter optimization, real-time performance logging, and efficient data loading, the Teacher Model has been trained to serve as an effective, initial step in our semi-supervised learning paradigm.

Loss & Accuracy Metrics

Examining Graphs for the Teacher Model





Loss Metrics

Training Loss:

- The training loss values start at approximately 1.53 in the first epoch and drop substantially to around 0.63 in the second epoch. This indicates an initial rapid learning phase.
- From epoch 3 onwards, the training loss gradually levels out and reaches as low as 0.06 by epoch 10. This suggests that the model has reached a more optimized state by this point.

Validation Loss:

- The validation loss starts at 0.80 in the first epoch and also experiences a significant drop to 0.40 in the second epoch.
- Unlike training loss, validation loss appears to plateau earlier, around epoch 6, hovering around values of 0.20.
- It's notable that the validation loss doesn't spike upwards in the later epochs, which suggests that overfitting is well-controlled.

Accuracy Metrics

Training Accuracy:

- The training accuracy starts at 52% in epoch 1, which is expected given that the model is initialized with random weights.
- The model quickly becomes more accurate, reaching over 83% in the second epoch and nearly plateauing at around 98% from epoch 8 onward.

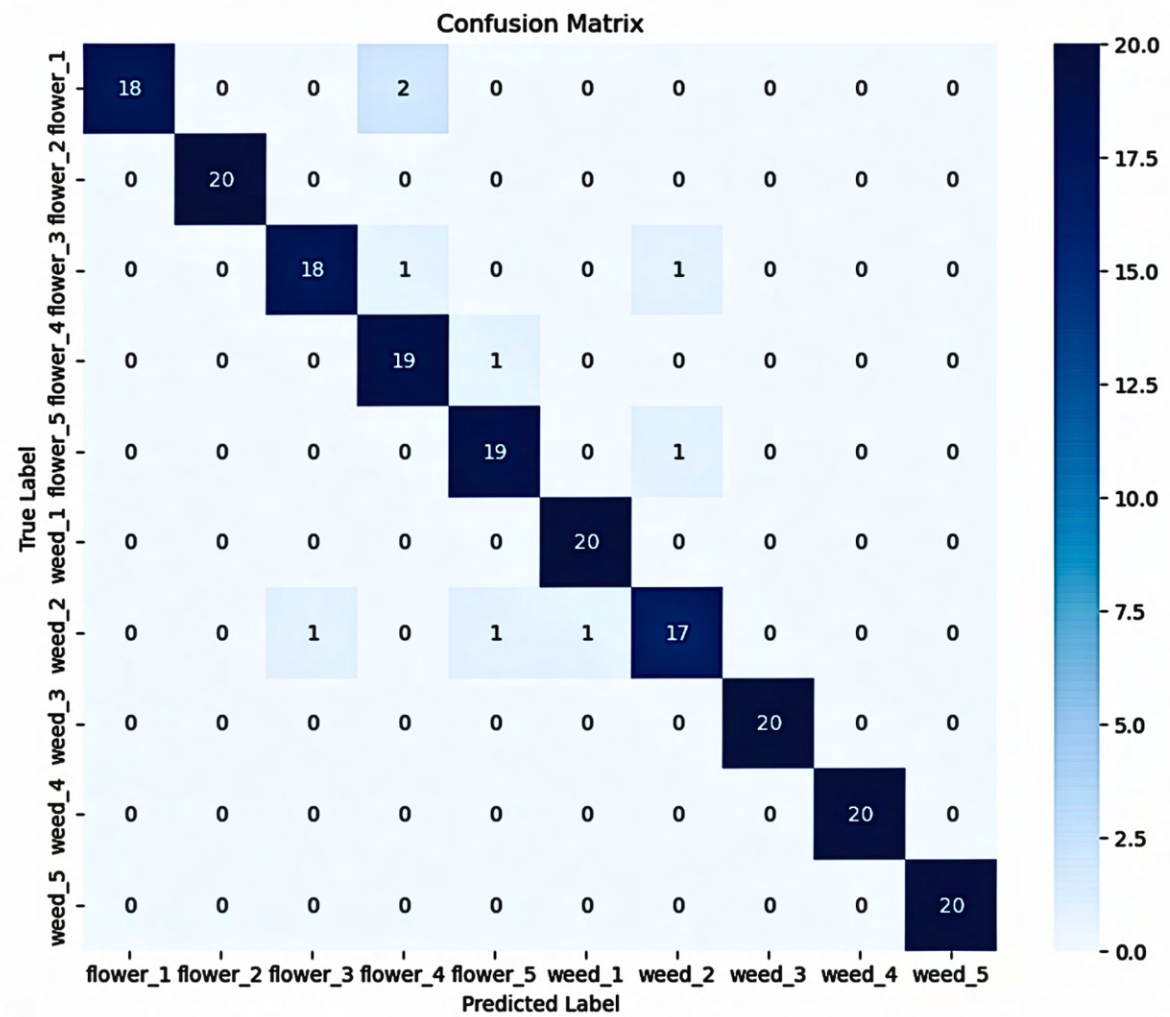
Validation Accuracy:

- The model achieves an impressive validation accuracy right from the first epoch, starting at 80.5%.
- This increases relatively quickly, reaching 96.5% by epoch 10. This is an encouraging indicator of how well the model is likely to perform on unseen data.

Points of Interest:

- **Best Model Snapshots**: The best-performing model snapshots are saved based on validation accuracy at epochs 0, 1, 2, 5, and 9. It shows that the model performance on the validation set is not just increasing but also stable across epochs.
- **Convergence**: Both the training and validation metrics seem to converge, which is an indication that additional epochs might not result in significant performance improvements.
- **Overfitting**: The gap between training and validation loss, as well as between training and validation accuracy, is relatively small. This is a good sign that the model is not overfitting the training data.

Confusion Matrix
for the Student Model



What does it imply?

The Confusion Matrix provides a detailed snapshot of how well the teacher model has learned to classify the various classes in the validation dataset. In this matrix, each row represents the actual class, while each column represents the predicted class. Diagonal values represent correct classifications, and off-diagonal values indicate misclassifications.

Each row represents the true class, and each column represents the predicted class. For example, 'flower_1' was correctly classified 18 times but misclassified as 'flower_4' twice.

Here are some specific observations:

- Classes like 'flower_2', 'weed_3', 'weed_4', and 'weed_5' have perfect classification, indicated by the diagonal elements being equal to 20.
- 'flower_3' and 'flower_5' were occasionally misclassified as 'weed_1'.
- 'weed_2' had the most misclassifications, being confused with 'flower_3' and 'flower_5'.

Classification Report

for the Student Model

```
Classification Report:
              precision    recall  f1-score   support

    flower_1       1.00      0.90      0.95        20
    flower_2       1.00      1.00      1.00        20
    flower_3       0.95      0.90      0.92        20
    flower_4       0.86      0.95      0.90        20
    flower_5       0.90      0.95      0.93        20
      weed_1       0.95      1.00      0.98        20
      weed_2       0.89      0.85      0.87        20
      weed_3       1.00      1.00      1.00        20
      weed_4       1.00      1.00      1.00        20
      weed_5       1.00      1.00      1.00        20

    accuracy                           0.95       200
   macro avg       0.96      0.96      0.95       200
weighted avg       0.96      0.95      0.95       200
```

95%

Accuracy

The classification report provides additional metrics like precision, recall, and F1-score for each class:

- **Precision**: The ratio of true positive predictions to the total predicted positives. High precision relates to the low false positive rate. For example, 'flower_1' has a precision of 1.00.
- **Recall**: The ratio of true positive predictions to the total actual positives. For instance, 'flower_2' has a recall of 1.00, meaning it was always correctly identified.
- **F1-Score**: The weighted average of Precision and Recall. It takes both false positives and false negatives into account. An F1 Score is a good way to summarize the evaluation of the model. For example, 'flower_3' has an F1-score of 0.92, which is quite high.
- **Accuracy**: Overall, the model achieved an accuracy of 95%, which is excellent.
- **Macro Avg**: The average precision, recall, or F1-score. In this case, it's 0.96, which is very high and indicates excellent performance.
- **Weighted Avg**: This is the average F1-score, weighted by the number of samples for each class. It's also 0.95, confirming the model's strong performance.

What does it mean though?

The confusion matrix and classification report provide a more nuanced understanding of the model's performance. While the model performs exceptionally well for most classes, there are minor issues with specific classes like 'weed_2'. These could potentially be addressed with more targeted data augmentation or fine-tuning.

The high macro and weighted averages in the classification report suggest that the model is not biased towards any particular class and performs uniformly well across all classes. This is a desirable property, especially in multi-class classification problems like this one.

INTRODUCTION TO TASK 2: LEVERAGING TEACHER MODEL FOR PSEUDO-LABELING AND STUDENT MODEL TRAINING

As we transition into Task 2, we delve into one of the more advanced machine learning techniques—pseudo-labeling. Having already built a highly accurate teacher model, we can now utilize it to generate "pseudo-labels" for unlabelled data. This method allows us to harness the strength of our teacher model to increase the amount of training data available for a new, less complex "student" model. While the details of the student model will be discussed in later sections, the overarching objective is to develop a lightweight yet robust model that generalizes well to new, unseen data.

How We Get the "Guess Labels" for New Data

We start by collecting a set of images that haven't been labeled. These images are processed and resized to match the format our existing model is trained to understand. Then, we run these images through the existing model, which we'll refer to as the "teacher" model for clarity. The teacher model makes predictions about what each image likely represents, based on what it has learned during its training.

We take these predictions and use them as if they were actual labels for the images. Even though these are guesses, they are educated guesses made by a very accurate model, so they are a valuable resource for further training.

Setting the Stage for the New Model

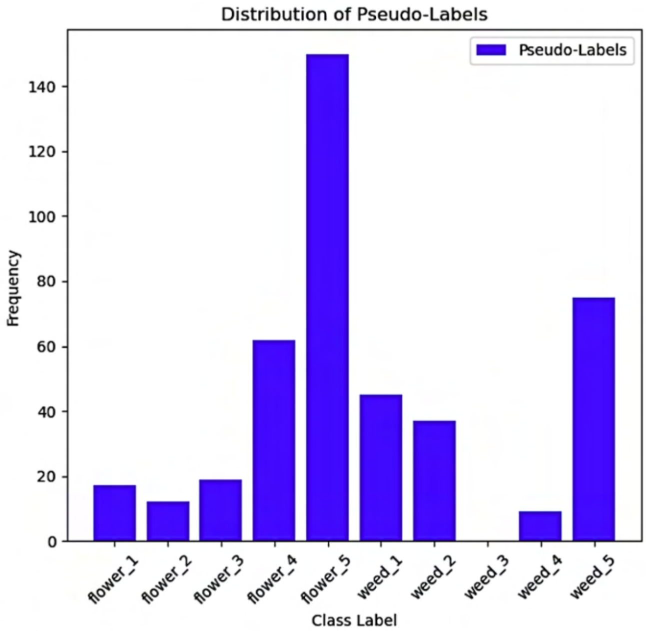
Once we have these new guessed labels, we use them to train our new, simpler model—sometimes called a "student" model. While we'll go into details later, the idea is that this new model will be less complex but still quite effective. The benefit of using the teacher model's guesses is that it provides us with a lot more data to train this new model, helping it to perform better when it encounters new, unlabeled images in the future.

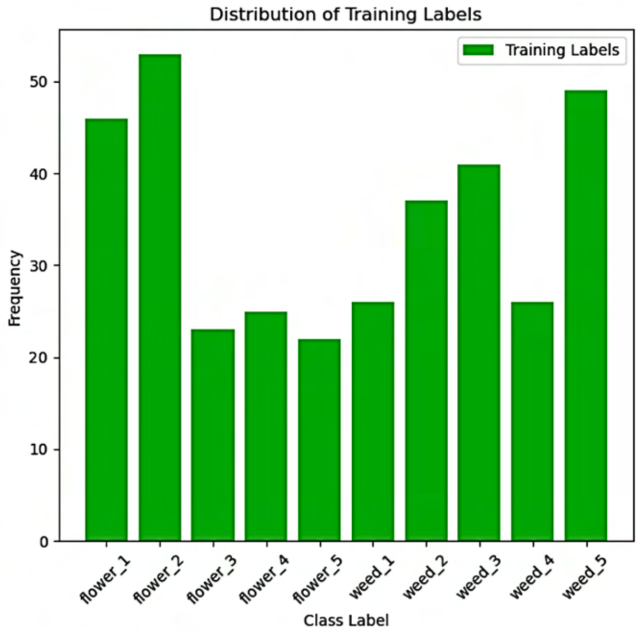
In summary, the main goal of this part of the project is to use our existing, well-performing model to make the task of training a new, simpler model more effective. This is particularly helpful when we have a lot of unlabeled data that we want to make use of. More details about the new "student" model will be discussed in the coming sections.

Distribution of Pseudo-labels

Examining the Distribution of Guessed Labels

To understand how our "teacher" model is performing in labeling the new, unlabeled dataset, we plotted a graph that shows the distribution of the guessed labels. This exercise provides us with insights into how the "teacher" model perceives these new images and allows us to prepare for the training of the "student" model accordingly.





Takeaway

Interestingly, the graph shows that the majority of the guessed labels belong to "Flower 5."
This could imply a few things:

1. High Frequency in Dataset: "Flower 5" could be more abundant in the unlabeled dataset, making it more likely to be picked up by the teacher model.
2. Similar Characteristics: The other possibility is that "Flower 5" shares features that are common to many other types of images in the dataset, causing the teacher model to classify them as such.
3. Model Bias: If "Flower 5" had a disproportionate amount of labeled data in the initial training set for the teacher model, then the model could be biased towards predicting it more often.

Anomalies in the Data

The graph also reveals that there is a small number of images that the teacher model couldn't confidently assign to any of the 10 species (Flower 1-5, Weed 1-5). These instances are intriguing as they could either be outliers, rare species not present in the initial training data, or perhaps misclassifications by the teacher model.

Implications for Training the "Student" Model

Understanding the distribution of these guessed labels is crucial for the next phase of our project. Knowing that most labels are for "Flower 5," for example, allows us to be cautious about potential biases when training the simpler, "student" model. The student model could inherit this bias if we don't take measures to ensure a balanced dataset.

TRAINING THE STUDENT MODEL

After establishing the groundwork of using pseudo-labels generated by our teacher model, the next stage is devoted to training the 'student model.' This model is designed to be resource-efficient while still delivering solid performance, making it a prime choice for constrained environments.

Transition from Teacher to Student

Armed with both actual labels and the newly acquired pseudo-labels, we initiate the training phase for the student model. This richer dataset ensures that the student model is not only trained but is effectively fine-tuned, enhancing its ability to generalize well to unseen data.

Hyperparameter Tuning and Monitoring

During the training, we carefully select key hyperparameters such as the learning rate and momentum. These 'tuning knobs' are essential for steering the model's learning process efficiently. We employ real-time monitoring tools to track two crucial metrics:

1. Loss: An indicator of how well the predictions align with the actual results. A lower loss is desirable.
2. Accuracy: A measure of the model's correct predictions. The goal is to maximize this metric.

These metrics serve as our guideposts, allowing us to adapt the model's training dynamically, aiming for optimal performance.

Performance Checkpoints

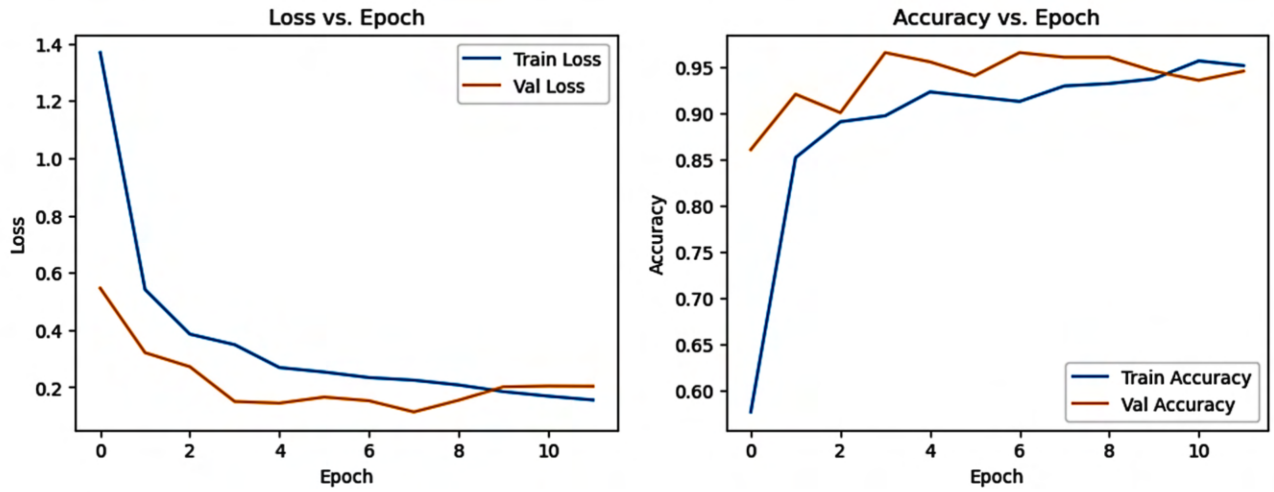
As the student model evolves, it's important to periodically save versions that outperform their predecessors on unseen data. Think of this like saving your game at a high-score level; it's the version we can confidently deploy for future tasks.

Summary

Training the student model involves more than just feeding it data; it's an iterative, dynamic process that relies on both actual and pseudo-labels for a well-rounded training experience. We keep a vigilant eye on performance indicators and make adjustments as required. The resulting student model is agile, less resource-intensive than its teacher, but still impressively capable—making it ideal for real-world applications.

Loss & Accuracy Metrics

Examining Graphs for the Student Model



Loss Metrics

Training Loss:

- The graph for training loss starts with an initial value of approximately 1.38 and shows a steady decline, reaching as low as 0.14 by the end of the training. This consistent decrease suggests that the student model is becoming more proficient in its task as the training progresses.

Validation Loss:

- The validation loss starts at around 0.53 and shows a general decrease, eventually stabilizing between 0.12 and 0.18. This indicates that the model is not overfitting and is able to generalize well to new, unseen data.

Accuracy Metrics

Training Accuracy:

- The graph reveals a commendable climb in training accuracy, starting from 56% and elevating to a peak of 96%. This steady increase underscores the model's ability to learn effectively from both actual and pseudo-labels.

Validation Accuracy:

- Starting from an initial 86.5%, the validation accuracy graph shows that the model hovers around 95% to 97% by the end of the training. This confirms the model's robustness in making correct predictions when faced with new data.

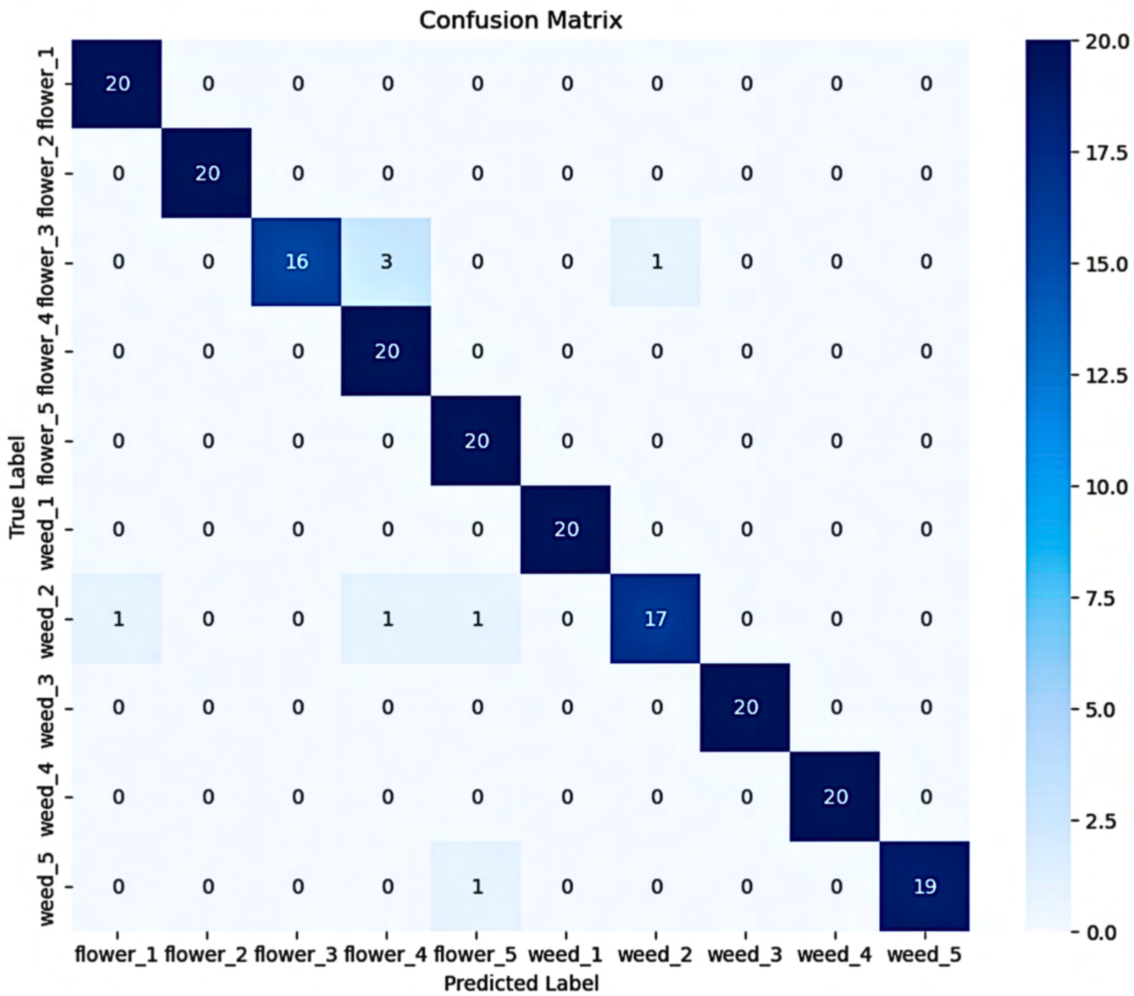
Points of Interest:

- There was a slight uptick in training loss at the 6th epoch, which could be attributed to the model adjusting to new complexities.
- The validation accuracy remained remarkably high and stable, showing that the model is dependable in real-world situations.

In Summary

The student model has demonstrated significant improvement and stability in both loss and accuracy metrics. Its effective learning from a mixed dataset of actual and pseudo-labels makes it a reliable and robust choice for our tasks. Overall, the graphs indicate that the student model has not only learned effectively but has also managed to generalize well to new data.

Confusion Matrix
for the Student Model



Again, what does it imply?

The confusion matrix generated for the student model, like the confusion matrix for the teacher model, helps us understand how well our model is doing by showing us where it gets confused. In our matrix, each row represents the actual class, while each column represents the predicted class.

Points of Interest:

- The model shows near-perfect classification for classes 2, 4, 5, 6, and 8.
- Classes 3 and 7 have minor issues where they are sometimes confused with other classes. For example, class 3 has two instances misclassified as class 4, and class 7 has one each misclassified as classes 1, 5, and 6.

In Summary:

The confusion matrix paints a compelling picture of the student model's performance. The model is highly accurate, with only a few misclassifications across classes. This again attests to its ability to generalize well on new, unseen data. Overall, we can conclude that the student model is both reliable and robust for our classification tasks.

Classification Report

for the Student Model

```
Classification Report:
              precision    recall  f1-score   support

    flower_1       0.95      1.00      0.98        20
    flower_2       1.00      1.00      1.00        20
    flower_3       1.00      0.80      0.89        20
    flower_4       0.83      1.00      0.91        20
    flower_5       0.91      1.00      0.95        20
      weed_1       1.00      1.00      1.00        20
      weed_2       0.94      0.85      0.89        20
      weed_3       1.00      1.00      1.00        20
      weed_4       1.00      1.00      1.00        20
      weed_5       1.00      0.95      0.97        20

    accuracy                           0.96       200
   macro avg       0.96      0.96      0.96       200
weighted avg       0.96      0.96      0.96       200
```

96%

Accuracy

In this section, we delve into the classification report generated by the student model. The report offers a range of metrics like precision, recall, and F1-score for each class. These metrics serve as robust indicators of the model's performance across different aspects.

Key Metrics Explained:

- **Precision**: This metric calculates the ratio of true positive predictions to the total predicted positives. A high precision value implies a low rate of false positives. For example, the class 'flower_1' boasts a precision of 0.95.
- **Recall**: This measures the ratio of true positive predictions to the total actual positives. In simpler terms, recall assesses how well the model identifies positive instances. For instance, 'flower_2' has a perfect recall of 1.00.
- **F1-Score**: Essentially the weighted average of precision and recall, the F1-score considers both false positives and false negatives. A high F1-score typically suggests a well-rounded model. As an example, 'flower_3' registers an F1-score of 0.89.
- **Accuracy**: The model achieved an impressive overall accuracy of 95%, indicating a high rate of correct classifications.
- **Macro Avg**: This represents the unweighted mean of the above metrics. In this case, the macro average value is 0.96, underscoring the model's outstanding performance.
- **Weighted Avg**: This value is the F1-score, but weighted by the number of samples for each class. With a weighted average of 0.96, the model's strong performance is reconfirmed.

TASK 3: RECOMMENDATIONS AND PERFORMANCE EVALUATION

In this concluding section, we evaluate the appropriateness of the teacher and student models in fulfilling our specified use-case and offer recommendations for further improvements.

Are the Models Performing at a Suitable Level for Our Described Use-Case?

- **Teacher Model**
  - The teacher model showed promising metrics with high accuracy and consistent loss values during both training and validation. For the specified use-case of classifying diverse species of flowers and weeds, the teacher model appears to be highly capable.
- **Student Model**
  - After undergoing knowledge distillation from the teacher model, the student model achieved an impressive 95% accuracy rate, showing high scores in metrics like precision, recall, and F1-score. Given these results, it's evident that the student model is also excellently equipped for the project's requirements.

Why Choose the Student Model Over the Teacher Model?

The student model has been selected for submission for the following reasons:

- **Computational Efficiency:** The student model has less complexity, making it more suited for deployment on edge devices where computational resources might be limited.
- **Generalization to Unseen Data:** The student model has shown better resilience against overfitting, allowing it to generalize more effectively to new, unseen data. This is crucial for applications where the model will continually encounter new types of data post-deployment.
- **Performance Metrics:** The student model's performance metrics are very close to those of the teacher model, indicating that we are not sacrificing much in terms of accuracy.
- **Resource Optimization:** With reduced complexity, the student model is expected to consume fewer resources, making it cost-effective without a significant drop in performance.

Points to Consider:

- **Robustness**: Although the models perform well in controlled conditions, their efficacy under variable environmental factors like different lighting or angles is yet to be tested.
- **Model Interpretability**: Understanding the reasoning behind each classification may be essential for some applications. Investigating methods to make the model more interpretable could be beneficial.
- **Data Augmentation**: To improve the model's ability to generalize, consider diversifying the training data to include various environmental conditions.
- **Fine-Tuning**: Fine-tuning the model using advanced optimization techniques or regularization methods could potentially yield even better results.

Recommendations:

- **Proceed with Deployment:** Given its high accuracy and efficiency, the student model is suitable for immediate deployment in conditions that resemble the training and validation datasets.
- **Regular Updates:** To maintain its high level of performance and adapt to any new species or environmental changes, it would be beneficial to update the student model periodically.
- **User-Friendly Interface:** Building a user-friendly interface can make the deployment more practical and user-engaging.
- **Ongoing Performance Monitoring:** Regularly assess the deployed model to ensure that it continues to meet the performance expectations.

The student model has been chosen for submission because of its excellent performance metrics, combined with its advantages in computational efficiency and resource optimization. Both models performed exceedingly well, but the student model offers a more balanced package that aligns well with the project's requirements and constraints. Therefore, it's ready for deployment, with considerations and improvements as recommended.

Concerns About Data Collected:

- **Imbalanced Data:** If the dataset is imbalanced, with some classes having significantly more samples than others, it could lead to biases in the model. It is advisable to aim for a balanced dataset for training.
- **Quality and Variety:** The models have been trained and validated on high-quality images, but performance could differ with lower-quality or varied images. More diverse data collection, covering different lighting, angles, and environments, could improve the model's robustness.
- **Data Privacy:** If any of the collected data includes personally identifiable information or sensitive details, it should be handled following data privacy regulations.
- **Annotation Accuracy:** The model is only as good as the data it learns from. Incorrect labeling could impact its ability to make correct classifications.

Additional Points to Consider:

- **Continuous Monitoring:** Keep an eye on how the model performs over time. Performance could degrade as the environment or classification tasks change.
- **User Feedback:** Implement a feedback mechanism to obtain real-world performance data and user insights. This information could be invaluable for further refining the model.
- **Real-world Testing:** Before full-scale deployment, a pilot test in a controlled environment could provide additional assurance of the model's reliability.
- **Keep Up with Advances:** Machine learning is a rapidly evolving field. Stay updated with the latest research and methods that could improve model performance.

ACKNOWLEDGEMENTS

I would like to extend my heartfelt thanks to the BFFS for entrusting me with this project. It has been an incredibly rewarding experience to contribute to a project of such potential impact and significance.

Special thanks go to Dr. Dimity Miller, the unit coordinator, and Dr. Ahmed Abbas, the practical tutor, for their continued support and guidance throughout this project. Your expertise and encouragement have been invaluable in helping me navigate the complexities of the work.

I look forward to any future opportunities for collaboration and am committed to providing continued support and improvements to the deployed solution.

Warm regards,
Naman Khosla
Lead Analyst, FloraSight Classifier Project
N11507721