Predicting Plant Traits with Ensemble Machine Learning Methods - CNN's vs Transformers

Jaimil Dalwadi
School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1
jaimil.dalwadi@uwaterloo.ca
August 13, 2024

Abstract

This report presents a comparative study of two different ensemble models used to predict plant traits from images and ancillary data. The first model trained and evaluated was a InceptionResNetV2-based CNN for images paired with a simple 2-layer neural net to regress tabular ancillary data. The second model was similar but was based on a Vision Transformer (ViT) with pretrained weights rather than the CNN for images, but the same neural net for the ancillary data. Effects of data augmentation, normalization, and model complexity were tested. The ViT transformer-based model paired with randomized image augmentation, normalized inputs/outputs, and simpler layers performed better when considering R2 score on the test set. The code used in this project is available here.

1 Introduction

In recent years, AI has become more prominent in a variety of tasks, one which is predicting plant traits automatically from photographs and data about each plant regarding local climate, soil, and satellite information. I was given 164 columns of ancillary data in csv files and images (128 by 128) with 43,363 samples to train with and needed to predict 6 traits for 6,391 samples in the test set. A study suggested that with AI, predicting these traits is possible (Schiller et al., 2021) and this research project aims to test that hypothesis through different models and techniques. These includes normalization, data augmentation, and model complexity. A summary of findings is shown below in table 1, but examined further throughout this report.

Table 1: Average R^2 Scores for Different Model Variations

Model	Technique	Before Kaggle R^2 Score	After Kaggle R^2 Score
CNN (Inception)	Data Normalization	-11549.5018	0.08520
	Image Augmentation	0.1021	0.1214
Transformers (ViT)	Image Augmentation	0.28662	0.29070
	Model Complexity Reduction	0.271225	0.3061

2 Related Works

After scouring the internet, I found several similar competitions on Kaggle and the performance of people who take similar approaches (a combined CNN and regression model). For example, in the referenced competition – PlantTraits2024-FGVC11 (Kaggle 2024), two approaches included:

Submitted to CS480/680, 2024 Spring. Do not distribute.

- EfficientNetV2 based CNN paired with a dense neural net from the Keras team (https://www.kaggle.com/code/awsaf49/planttraits2024-kerascv-starter-notebook). This has an R2 score of approx. 0.05.
- A complex, 3 headed approach with a Vit backbone for images and additional pre-processing to extract species of each plant resulting in an R2 score of approx. 0.6. (https://www.kaggle.com/competitions/planttraits2024/discussion/510393)

There were some stark differences in the data used for this research project compared to the referenced competition. The other competition required standard deviation for each output as well as the existing mean outputs mentioned in my project. They had a larger image dataset (62,000 samples), and over 300 columns of data. Due to the limited time, compute power, and data I had for this project, I decided to test a simplified version of the 3 headed model - simply using the ViT transformer with pre-trained weights for images and pairing it with the tabular data. I also decided to test a different base CNN – InceptionResNetV2 which is popular (Anwar 2020). Recently, ViT has been gaining traction by treating image patches as tokens, so it was of interest to test (Ranftl 2021).

3 Main Results

3.1 Problem Formulation

This problem can be explained as follows. Given an image I and ancillary data A for a specific ID, our task is to predict the vector of traits $\hat{Y} = [\hat{Y}_{1i}, \hat{Y}_{2i}, \ldots, \hat{Y}_{6i}]$ where each $\hat{Y}_{ji}$ corresponds to one of 6 plant traits for i'th sample. The actual/true values are denoted by Y_{ji}.

We want a model that can maximize the R^2 score, which is defined as:

$$R^2_{avg} = \frac{1}{6}\sum_{j=1}^{6}(1 - \frac{\sum_{i=1}^{n}\|Y_{ji} - \hat{Y}_{ji}\|^2}{\sum_{i=1}^{n}\|Y_{ji} - \bar{Y_{ji}}\|^2})$$

where n is the number of samples in the dataset, Y_{ji} is the true value of the i-th sample, $\hat{Y}_{ji}$ is the predicted value of the i-th sample, $\bar{Y_i}$ is the mean of the actual/true values. The R^2_{avg} score measures the proportion of variance of each predicted trait (averaged across 6 traits) that can be explained by my model.

3.2 Model Definitions

Before discussing various experiments, below are the two model definitions with a lack of detail due to popularity.

- **Model 1: InceptionResNetV2 with Linear Regression on Ancillary Data.** I used the InceptionResNetV2 architecture with custom training for image feature extraction. The extracted features were then combined with features from a linear regression model trained on the ancillary data.
- **Model 2: Vision Transformer (ViT) with Linear Regression on Ancillary Data.** The ViT architecture with pretrained weights from Google was used for image processing, paired with a linear regression model for the ancillary data.

Both the transformer and CNN outputs in their respective model structures were concatenated to the Ancillary part of the model (a simple Linear Layer from 164 inputs to 128, and again to 64). The concatenated features were applied a ReLU activation, Dropout of 0.5, and Linear layer to return 6 outputs.

The models were trained to minimize the MSE Loss score, in turn improving the R2 as well, using the Adam optimizer with a learning rate of $1e-4$.

3.3 Experimental Results

First off, the InceptionResnetV2 Model was constructed and tested with no techniques. The training dataset was randomly split with 80-20 split for train/validation.

3.3.1 Data Normalization

Upon submitting predictions, it was clear that there was something very wrong in my code. Thanks to a hint from Saber, I used the normalization techniques referenced in the paper (Schiller et al., 2021) to normalize the data. This led to a significant improvement in the R2 score and accuracy. The method of normalization was $\frac{X_i - X_{min}}{X_{max} - X_{min}}$ where the max and min come from the corresponding column in the training dataset. All input columns and output columns were normalized for training, and then reversed after predictions. Due to the stark differences seen in Figure 1, normalized data was used for all model variations going forward (unnormalized data was not tested for the transformers model).

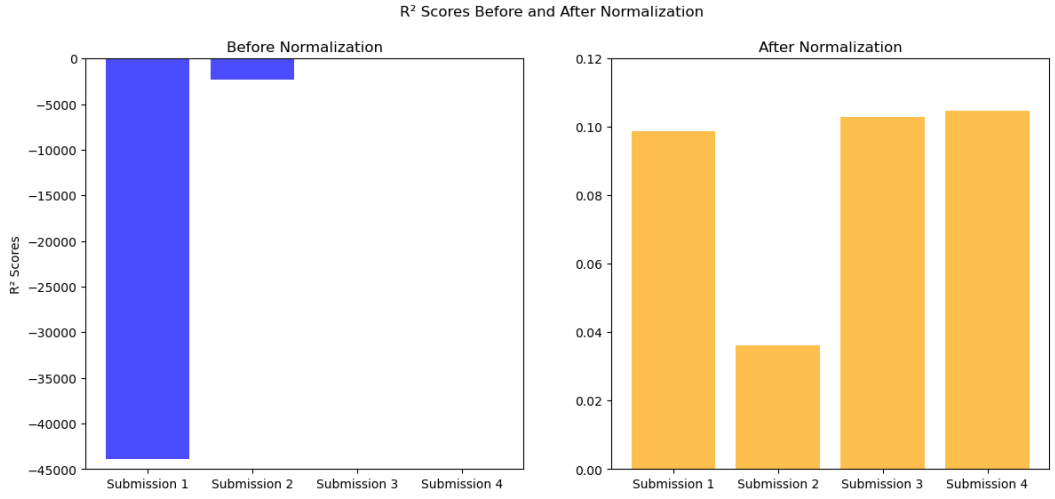

Figure 1: A side by side bar plot showing the change in R2 scores before and after normalizing the data.

3.3.2 Model Performance Comparison

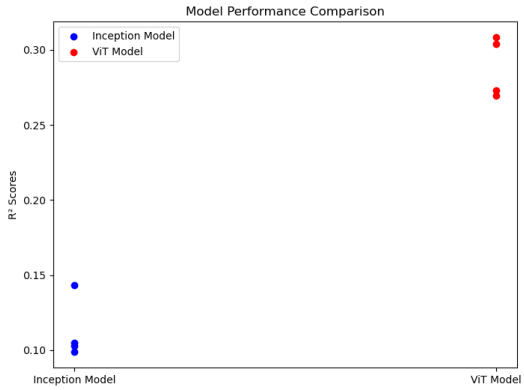

Figure 2: Scatter plot of R2 scores for each model type

After constructing and testing the ViT-based transformers model, it was very clear that it heavily outperformed the Inception-based CNN model (keeping the ancillary model consistent). It did take approx. 40 minutes to train per epoch compared to the 3 minute train time per epoch for the CNN model. The trade-off in training time was definitely worth the value.

3.3.3 Effect of Data Augmentation

A common technique, image augmentation was applied to the training datasets for both models. With a probability of 0.2, each image was randomly flipped (vertical/horizontal), contrast/brightness

were slightly modified (0.9 to 1.1), and all pixels were clipped to [0,1]. As a result, we see that the augmented scores on average are higher than those model scores that were trained with no augmentation as shown in Figure 3.

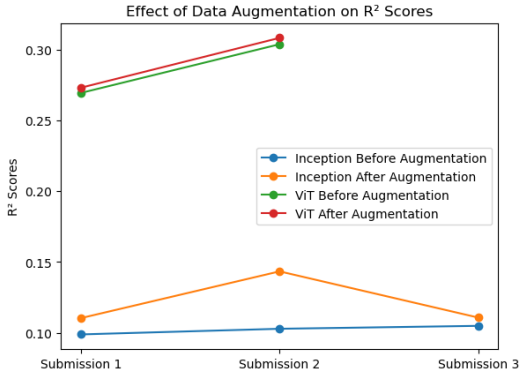

Figure 3: Plot of R2 scores for various submissions before and after augmentation for the two models.

3.3.4 Effect of Model Complexity

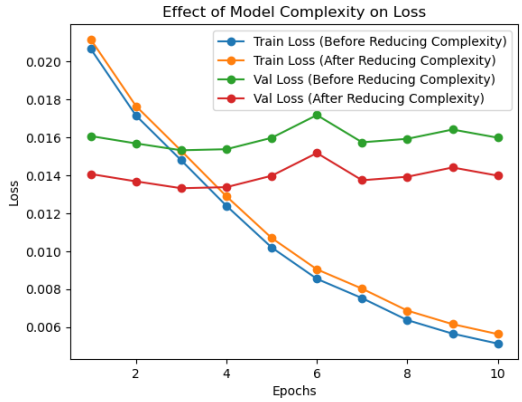

Figure 4: Comparison of MSE loss before and after reducing complexity throughout training.

Reducing model complexity simply refers to the fact that I removed an additional layer in the neural net for ancillary data (instead of having 163->128->64, I had 163->128). The outputs were then combined with outputs from the CNN/transformer to a concatenated linear layer and further down again to 6 for each target. This reduction in complexity shows a slight improvement in the model's performance in both transformer based and CNN based models as seen in Figure 4.

4 Conclusion

From my experiments, I observed that the ViT model generally outperformed the InceptionResNetV2 model, and image augmentation generally improved the performance of a model, more complex models weren't always better, and data pre-processing significantly helped improve model performance.

If I were to continue, it is worth exploring the ViT model further but training with custom weights. Another option is to use better methods for the ancillary data. This includes boosting the ancillary part of the model, bagging, or other methods for improved accuracy. These are a subset of a variety of other techniques for tabular data that were not tested due to time constraints. Testing different model types could also be fruitful, such as MLP regressors or XGboost for the ancillary data.

Acknowledgement

Thanks to the TA, Saber for providing various hints including using data normalization and augmentation. Giving credit to various LLM's online that assisted in writing code for image augmentation, normalization, and plotting. I rewrote and understood any code provided by LLM's.

References

Anwar (2020). "Difference between AlexNet, VGGNet, ResNet, and Inception". https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96. Accessed: 2024-08-11.

Kaggle (2024). "PlantTraits2024 - FGVC11". https://www.kaggle.com/competitions/planttraits2024/overview. Accessed: 2024-08-11.

Ranftl, R., A. Bochkovskiy, and V. Koltun (2021). "Vision Transformers for Dense Prediction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12179–12188.

Schiller, C., S. Schmidtlein, C. Boonman, A. Moreno-Martínez, and T. Kattenborn (2021). "Deep learning and citizen science enable automated plant trait predictions from photographs". *Scientific Reports*, vol. 11, no. 1, p. 16395.