CatBoost is all you need for Plant Traits Prediction

Chong Hou Choi
School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1
chchoi@uwaterloo.ca

Abstract

In this report, we present a method for addressing a Kaggle task focused on plant traits prediction, a multi-label prediction challenge. To tackle this task, we employ feature extraction from images using the open-source, pretrained DINOv2(Darcet et al. 2024) model by Meta. The resulting image embeddings are then concatenated with the tabular features. For the final prediction model, we use CatBoost(Prokhorenkova et al. 2019), chosen for its superior performance in handling categorical features. The code is in the following link: https://github.com/RainCCH/CS680/tree/main/proj

1 Introduction

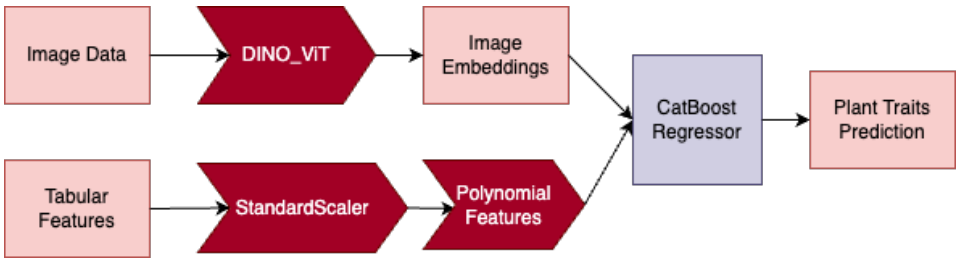

Figure 1: Pipeline of our model

Predicting plant traits is a challenging task that involves combining different types of data, managing categorical variables, and handling the complexity of predicting multiple traits simultaneously. In this report, we address these challenges using a method developed for a Kaggle competition on plant trait prediction. Our approach integrates image and tabular data, leverages a powerful feature extraction technique, and uses a machine learning model well-suited for this task.

1. **Fusion of Image Data and Tabular Data:**
   Plant traits can be predicted more accurately by combining information from images and tabular data. Images provide detailed visual information, while tabular data includes important structured information like genotype and environmental factors. We use the DINOv2 model by Meta to extract features from images and then combine these with tabular data, creating a comprehensive dataset that improves prediction accuracy.
2. **Categorical Data Handling:**
   Handling categorical data is crucial for accurate prediction. Traditional models often struggle with this, but we use CatBoost, a machine learning algorithm that is good at dealing with categorical variables. This choice helps capture important patterns in the data, leading to better predictions.

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

3. **Multi-Label Prediction:**
   Predicting multiple plant traits at once adds complexity, as these traits may be related. Our method effectively manages this by combining image features and tabular data, processed through CatBoost. We trained an independent CatBoost/XGBoost model for each of the trait, instead of training one model for all traits. This increases complexity but brings higher prediction accuracy.

In summary, our method addresses the key challenges in plant trait prediction by combining advanced feature extraction with a machine learning model that handles both categorical data and multi-label prediction. This integrated approach enhances the accuracy and reliability of predicting plant traits.

2 Related Works

2.1 Feature Extraction from Images

Feature extraction from images has traditionally relied on hand-crafted methods like Scale-Invariant Feature Transform (SIFT), which detects and describes local features, and Histogram of Oriented Gradients (HOG), which captures gradient orientation distributions for tasks like object detection. While effective, these methods often require manual tuning and may miss subtle details important in complex tasks such as plant trait prediction. With advancements in deep learning, convolutional neural networks (CNNs) have become the preferred approach, providing richer, more detailed image embeddings that better capture the essential features for specific tasks.

Recently, self-supervised learning models like DINOv2 have further advanced feature extraction by using the Vision Transformer (ViT) architecture. DINOv2 generates high-quality image embeddings without the need for labeled data, making it ideal for applications like plant trait prediction, where labeled datasets can be limited. Its ability to capture global context and subtle variations in images ensures that it produces expressive features suitable for complex multi-label tasks.

2.2 Tabular Features

For the tabular features, we first applied StandardScaler for normalization. We then applied PolynomialFeatures from Sklearn to transform the data in the preprocessing stage. There are several advantages using PolynomialFeatures:

1. **Interaction Between Features:** It creates interaction terms which can capture how the combination of two or more features influences the target variable. This helps capture the non-linear relationships between different features.
2. **Better Performances with Boosting models:** Models like XGBoost(Chen and Guestrin 2016) and CatBoost can benefit from polynomial features because these models can capture complex feature interactions. The polynomial features can help these models find more patterns in the data.

2.3 CatBoost vs XGBoost

CatBoost and XGBoost are both popular machine learning algorithms that belong to the family of gradient boosting techniques. They are widely used for tabular data and are known for their ability to model complex relationships in data with high accuracy.

In this task, CatBoost's performance is better than XGBoost due to the following reasons:

1. **Categorical Features:** CatBoost natively handles categorical features using ordered target encoding, eliminating the need for manual preprocessing like one-hot encoding, which leads to more efficient and accurate modeling, especially in datasets with many categorical variables.
2. **Overfitting Prevention:** CatBoost's ordered boosting technique reduces the risk of overfitting, which is particularly important in multi-label regression tasks where the model needs to generalize well across multiple targets.

3. **Integration with Image Data:** CatBoost can combine features extracted from images with categorical and numerical data, providing a solution that leverages the strengths of different data types for better overall performance.

3 Main Results

As shown in fig. 1, we first perform feature extraction on the images to transform them into image embeddings. For the feature extraction phase, we used different size of Vision Transformers provided by DinoV2, including ViT-Small, ViT-Base, ViT-Large and ViT-Giant. In this section, we will compare the performances of all these models.

Fixing all the hyperparameters of CatBoost, we can compare the performances of models using different feature extractors.

We will also compare the performances of CatBoost and XGBoost. In this experiment, we will fix all the other hyperparameters including the number of iterations, so that we can know which one is better for this task. From fig. 2, we know that CatBoost outperforms XGBoost in both ViT-Base and ViT-Large features.

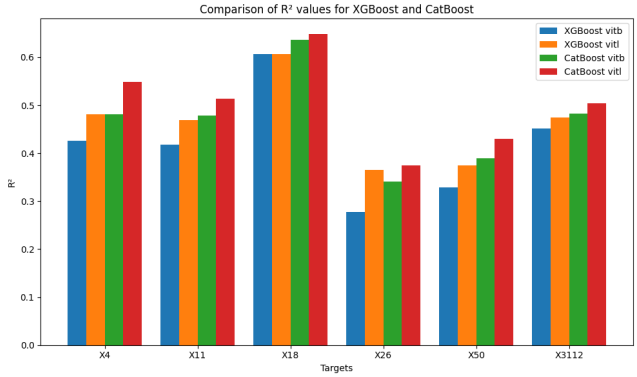

Figure 2: Comparison of R^2 values for XGBoost and CatBoost(on validation set)

The size of feature extractor also matters for the model. From table 1 we know that the larger the pretrained model is, the better the result will be.

Model	X4	X11	X18	X26	X50	X3112	Mean R²
ViT-Small	0.415	0.425	0.580	0.298	0.319	0.443	0.413
ViT-Base	0.481	0.479	0.636	0.341	0.389	0.482	0.468
ViT-Large	0.548	0.514	**0.648**	**0.375**	0.430	0.504	0.503
ViT-Giant	**0.553**	**0.531**	0.638	0.354	**0.442**	**0.514**	**0.505**

Table 1: R² values and mean R² for different ViT sizes at 1500 iterations.

There are several hyperparameters for us to adjust for better performances within the CatBoost model, including the number of iterations, learning rate and loss function. We apply RMSE(Root Mean Square Error) as the loss function and we will do comparisons to show how the number of iterations, learning rate and the choice of feature extractor affect the result.

From table 2 and fig. 4 we know that more iterations bring better result. However, we believe that the model will overfit at some point. However, due to the lack of computational resources, we are not able to test the limit of this.

4 Conclusion

In this report, we have proposed a method, using DinoV2's pretrained model as feature extractor of image data, and CatBoost as the model to handle both the image data and tabular data. This method

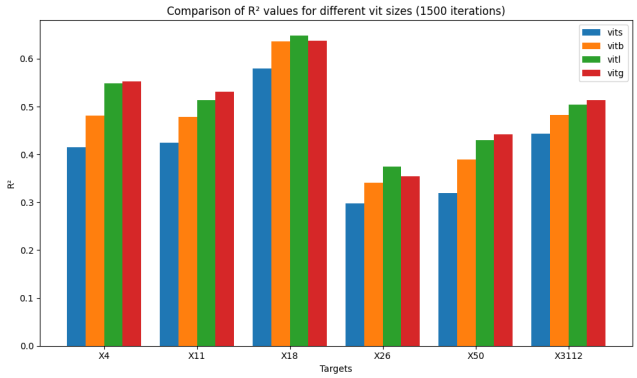

Figure 3: Comparison of R^2 values for different ViT sizes (1500 iterations)

Iterations	**R²**						**Mean R²**
	X4	**X11**	**X18**	**X26**	**X50**	**X3112**	
1000	0.545	0.508	0.647	0.375	0.423	0.500	0.500
1500	0.548	0.514	0.648	0.375	0.430	0.504	0.503
2000	0.552	0.518	0.649	**0.376**	0.432	0.505	0.505
2500	0.555	0.522	0.650	0.374	0.436	0.507	0.507
3000	**0.556**	**0.524**	**0.651**	0.372	**0.436**	**0.508**	**0.508**

Table 2: R^2 values for each target across different iterations on validation set.

is effective on this plant traits prediction task, leading to a 10% increase compared to the baseline model. However, there are still several ways to explore to improve the model's accuracy.

1. **Interaction Between Traits:** Instead of training a CatBoost Regressor model for each trait, there might be some relationship between traits, which means that the prediction of one trait might be valuable for the prediction of another trait. Therefore, we can try to add the prediction of the previous traits as new features for other traits.
2. **More Iterations:** As shown in table 2, even for 3000 iterations, there is no signal that the model is overfitting, as the accuracies for most traits are still increasing, except X26. However, due to the limit of computation resources, the experiment is not complete.

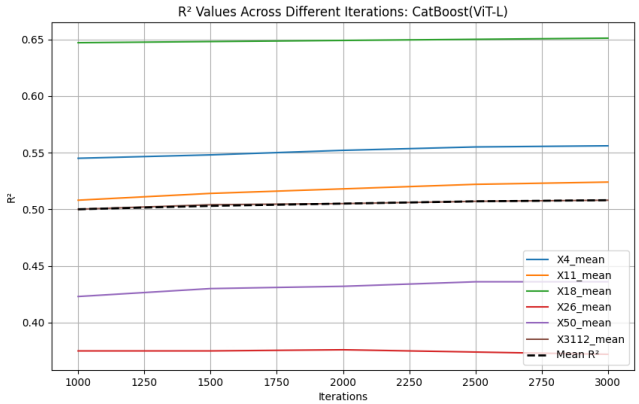

Figure 4: R^2 Values Across Different Iterations

References

Chen, T. and C. Guestrin (Aug. 2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM.

Darcet, T., M. Oquab, J. Mairal, and P. Bojanowski (2024). "Vision Transformers Need Registers". arXiv: 2309.16588 [cs.CV].

Prokhorenkova, L., G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin (2019). "CatBoost: unbiased boosting with categorical features". arXiv: 1706.09516 [cs.LG].