

CSC 542 - Neural Networks

Plant Trait Prediction



TEAM 68

Satyajeet Patil
Amitesh Patil
Parth Kulkarni

NC STATE UNIVERSITY

Introduction

- This project aims to **predict plant properties** - from citizen science plant images and supporting geographical data
- Plant traits are crucial for understanding ecosystem dynamics.
- For eg:
 - canopy height indicate a plant's competitive ability for sunlight,
 - Leaf mass per leaf area highlight adaptations to wind or drought
- As conditions evolve, plants may adapt their traits or shift their distributions, leading to significant alterations in ecosystem functioning.
- Our inference tasks involve developing models that can analyze plant photographs to predict various traits accurately.

Problem Statement







- Using plant traits to assess ecosystem functioning can help **understand how environmental changes affect ecosystems.**
- A good tool for measuring traits could make monitoring ecosystems easier, especially since we expect traits to change due to climate change.
- By using over 50,000 labeled images and location data, we aim to predict plant traits.
- This will help us learn more about how plants adapt to changes, improving our understanding of global ecosystems.

Dataset

- Source of data is Kaggle. The data has been taken from the TRY database (trait information) and the iNaturalist database (citizen science plant photographs). The inputs are images (jpeg) tsv, and csv files.
- The desired outputs are predictions of six different traits -
 1. Stem specific density (SSD) or wood density (stem dry mass per stem fresh volume)
 2. Leaf area per leaf dry mass (specific leaf area)
 3. Plant height
 4. Seed dry mass
 5. Leaf nitrogen (N) content per leaf area
 6. Leaf area (in case of compound leaves: leaf, undefined if petiole in- or excluded)
- Number of images for training and testing
 - ❑ Train images- 55.5k files
 - ❑ Test images- 7133 files
- Size of the tabular data -
 - ❑ train.csv :- 55489, 176 (78.52mb)
 - ❑ test.csv:- 6545, 164 (8.21mb)

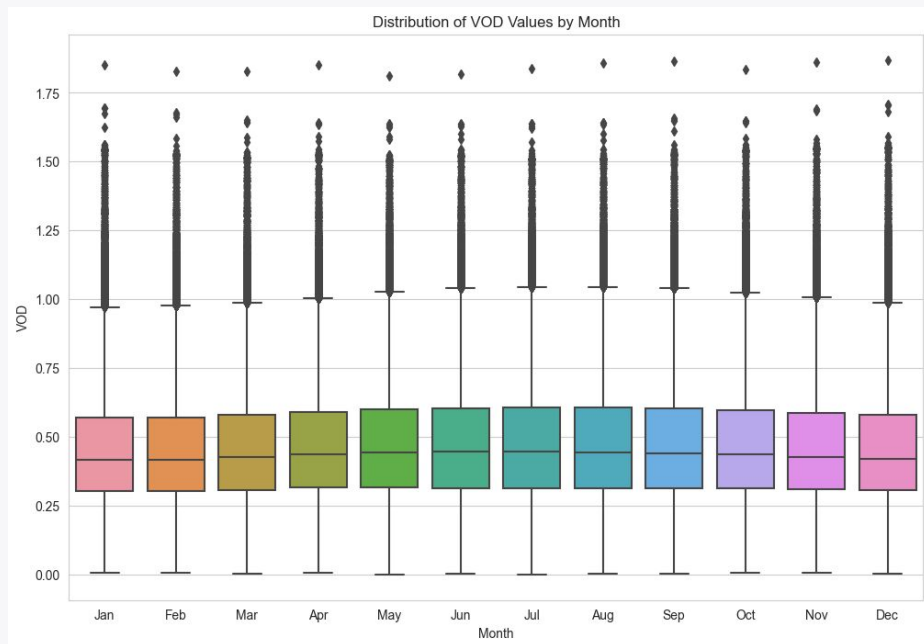
Data Explorer

3.45 GB

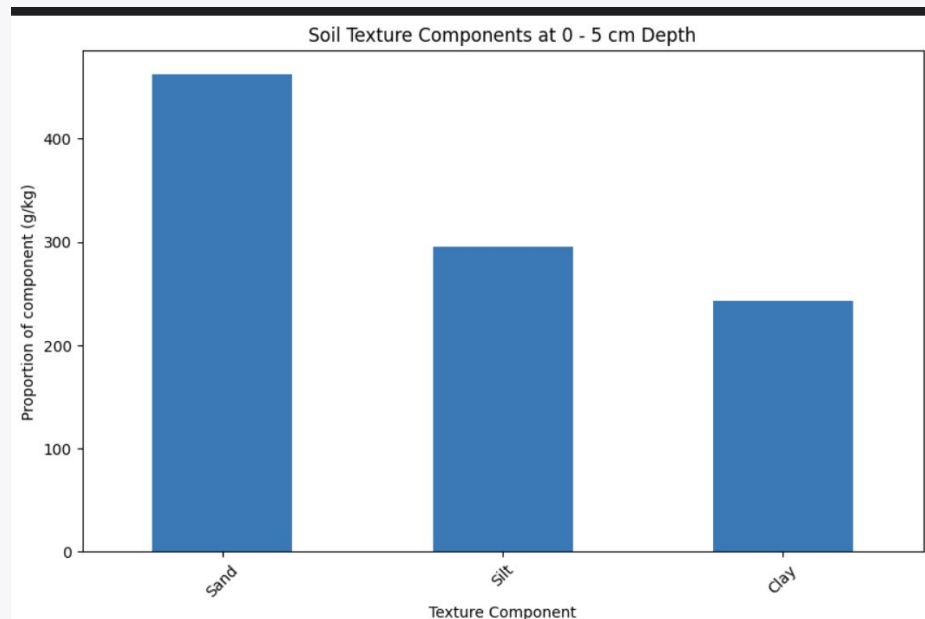
- ▶  test_images
- ▶  train_images
-  sample_submission.csv
-  target_name_meta.tsv
-  test.csv
-  train.csv



Exploratory Data Analysis

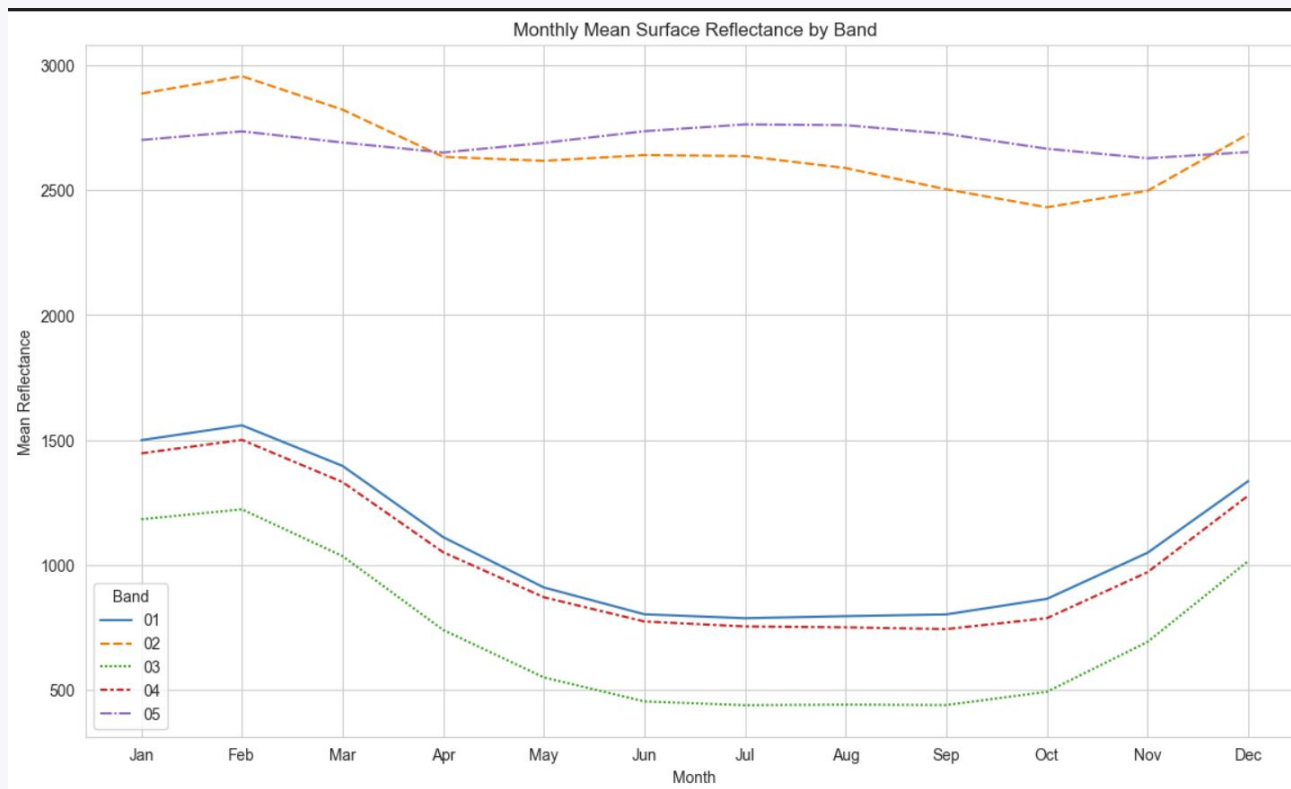


Distribution of VOD (data from radar that is sensitive to water content and biomass) values across months



Soil texture components across 0-5 cm depth

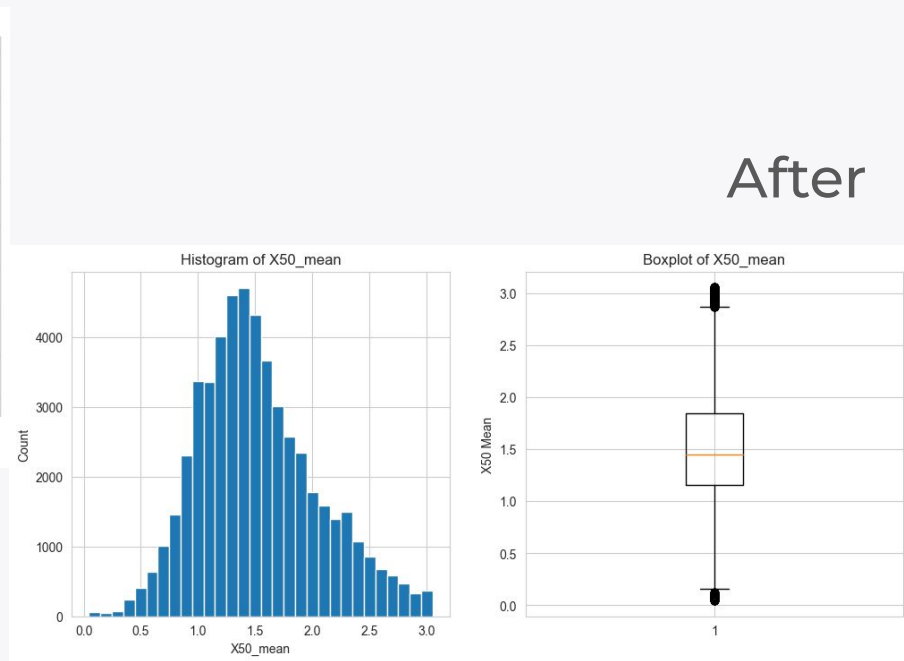
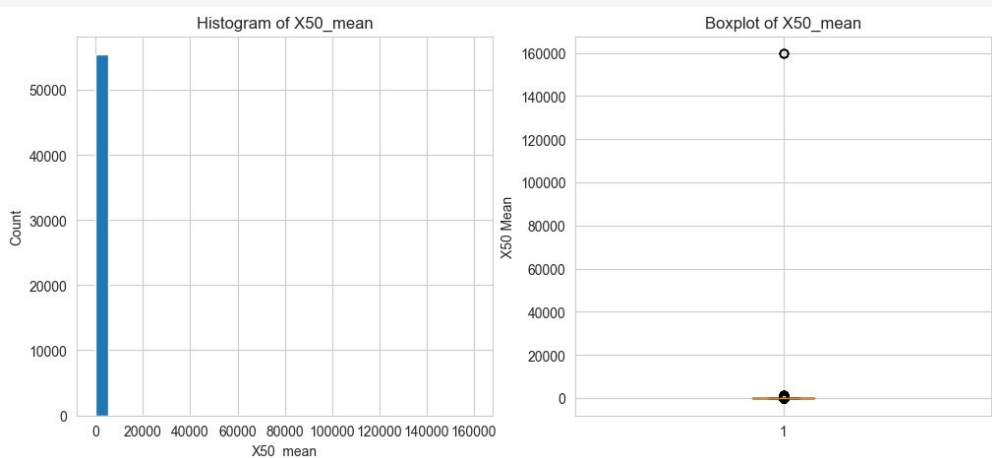
Exploratory Data Analysis



Monthly mean optical reflectance of sunlight for each band

Preprocessing

- Dropped columns having too many null values
- Grouped columns into four types:- WORLD CLIMATE, SOIL, MODIS, VOD
- Removed outliers using IQR method.
- **Image Preprocessing- flip, brightness, hue, saturation, crop**
- Normalized all the attributes.



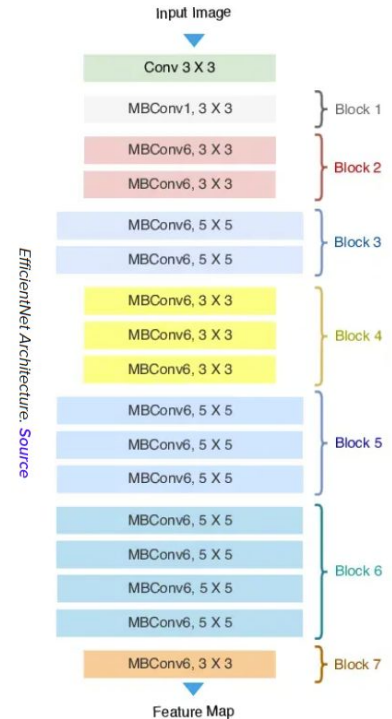
Before

MODELS

BASELINE:

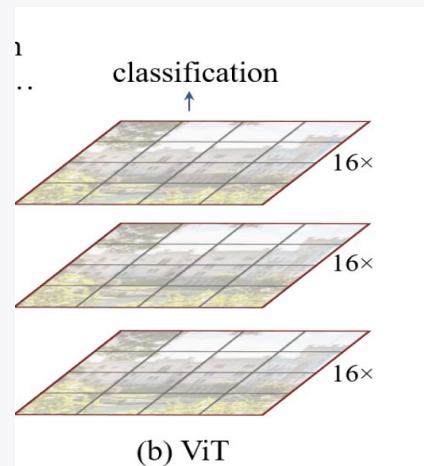
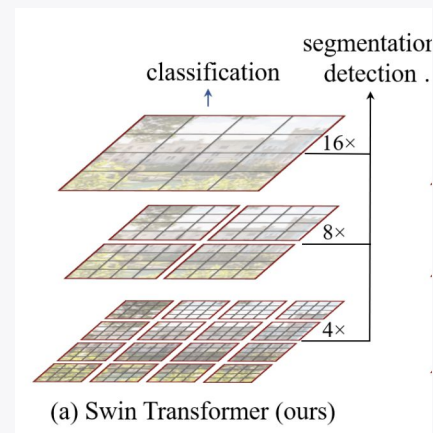
Efficient Net B0 with MLP

- CNN model trained on images from Imagenet database
- Uses Compound scaling method that scales the depth, width, resolution.
- The MBConv layer is a fundamental building block of the Efficient Net architecture.
- The MBConv layer starts with a depthwise convolution, (1x1 convolution) that expands the number of channels, and finally, followed by another 1x1 convolution that reduces the channels back to the original number. It allows the model to learn efficiently while maintaining a high degree of representational power.
- Image size (224, 224)
- The features are extracted from the Efficient Net B0 and concatenated with tabular data.
- The data is given to a **Multi layer perceptron** to get the prediction of the target variables.



Final Model: Swin Transformer with MLP

- Based on Vision Transformer Arch
 - Image divides into 16x16 patches
 - Patches are converted to Path Vectors through Linear Transformation
 - Each Path Vectors combine with Positional embeddings
- Problem with vision Transformer
 - Extraction of patches is the problem.
 - 256x256 px image can be divided into 16x16 patches -> 16 image tokens
 - But for HD images (1920x1920) -> 120 image tokens
 - Decent for one label tasks
 - In these cases 16x16 patches become very huge to extract smaller details
 - 256x256 has 63k tokens
 - Forget about 4k Images!
- SWIN Architecture
 - Image -> 4x4x3 channels -> 48 features -> Linearly transformed
 - Divide and Conquer: Attention is introduced but it doesn't consider all at once,
 - It slides and maintains attention for a fixed number of neighbouring sequences
 - This output is merged by encoder
 - Passed through Linear Projection to decrease dimensionality (e.g. 4c to 2c)
 - Steps 1 to 4 happen for multiple runs, where the window is "shifted"



Results

	Baseline Model	Final Model
R2 Score	0.26	0.71
Mean Absolute Error (MAE)	0.58	0.29
Mean Squared Error (MSE)	0.74	0.40

Conclusion

- Vision Transformers and derived models like SWIN transformers are capable of capturing complex information from image data.
- Architectures using SWIN transformers outperform others that use pretrained CNNs like efficient Net.





References

- Dataset -
<https://www.kaggle.com/competitions/planttraits2024/data>
- Swin Transformer: Hierarchical Vision Transformer using Shifted Windows.
arXiv:2103.14030 [cs.CV]
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.
arXiv:1905.11946 [cs.LG]



thank:
- you