Vision Transformers for Plant Traits Prediction

Lisa Fung & Annabelle Aurelia Jayadinata

AFFILIATIONS

Stanford University, Department of Computer Science & Electrical Engineering, CS231N

01. Problem

- **Climate change** is expected to cause approximately 250,000 additional deaths per year in 2030-2050, from under nutrition, malaria, diarrhea and heat stress alone.
- Rapidly transforming biosphere causes shifts in ecosystem species distributions and plant adaptations, measured by **changes in plant traits**
- Predicting the global-scale impact of climate change on ecosystems is **difficult to quantify** due to **insufficient data on plant traits**

Objective: Employ deep learning-based regression models, including Convolutional Neural Networks (CNNs) and Vision Transformers to predict plant traits from photographs and ancillary geodata

trait_ID	trait_name
X4	Stem specific density (SSD) or wood density (stem dry mass per stem fresh volume)
X11	Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA)
X18	Plant height
X26	Seed dry mass
X50	Leaf nitrogen (N) content per leaf area
X3112	Leaf area (in the case of compound leaves: leaf, undefined if petiole in- or excluded)

02. Background

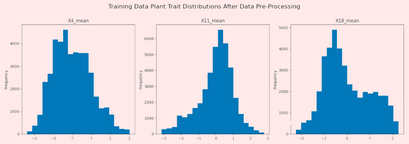
- **Datasets:** Data is a pre-labeled link between the **TRY database** (plant trait information) and the **iNaturalist database** (55,500 citizen science plant photographs) provided through Kaggle.
  - Additional location-based **geo-datasets** for each plant photograph, including temperature and precipitation data, and satellite data.
  - Use a 4:1 split for training and validation.
- **Performance Metrics:** The evaluation metric is the mean R2 (R-Squared) over all 6 traits.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

- **Loss Function:**

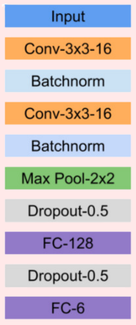
$$\text{SmoothL1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases}$$

- **Data Preprocessing:**
  - Filter outliers using a quantile range (0.005, 0.995) for all traits
  - Apply a logarithmic base 10 transformation to right-skewed trait data, which includes the five plant traits except trait X4
  - Normalize data to achieve a zero mean and unit standard deviation across all traits



03. Methods

BASELINE MODEL

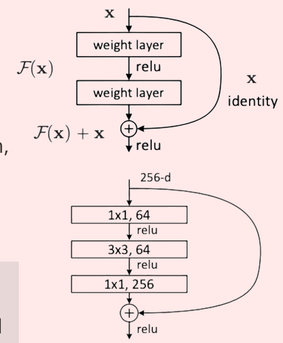


CNN model with two 3x3 convolutional layers, one 2x2 max pooling layer, and two fully connected layers.

- ReLU activations, batch normalization, and dropout. Training uses Adam optimizer with a step learning rate scheduler initialized to 1e-4.

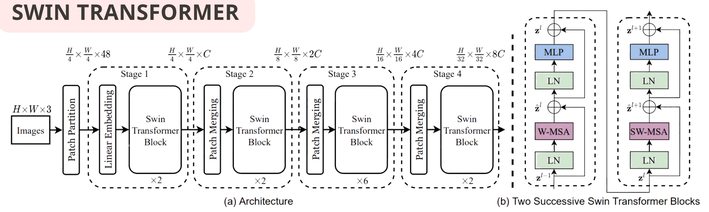
VGG-16 A simple CNN architecture consisting of 16 layers (13 convolutional and 3 fully connected layers)

RESNET-50



- ResNet's **residual mappings** allow the stacked nonlinear layers to fit another mapping of F (x) := H(x) − x. The original mapping is recast into F(x) + x.
- In ResNet-50, for each residual function F, we use a stack of 3 layers which are 1×1, 3×3, and 1×1 convolutions
  - 1×1 layers reduce and then increase (restore) dimensions
  - 3×3 layer is a bottleneck with smaller input and output dimensions

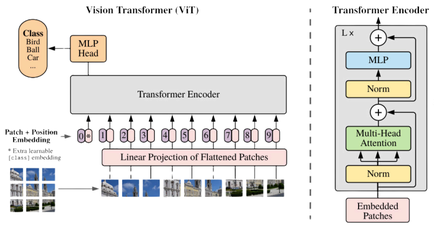
SWIN TRANSFORMER



(a) Architecture (b) Two Successive Swin Transformer Blocks

- Swin Transformers rely on **hierarchical feature extraction**
  - Features extracted at different scales, then combined to form a holistic representation of the image
  - At each level, features are split into non-overlapping patches and processed by a series of transformer blocks

VISION TRANSFORMER (VIT)



$$z_0 = [x_{\text{class}}; x_p^1 E; x_p^2 E; \ldots; x_p^N E] + E_{\text{pos}} \quad (1)$$

$$\text{where } E \in \mathbb{R}^{(P^2 \cdot C) \times D},\ E_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$$

$$z'_l = \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}, \quad l = 1, \ldots, L \quad (2)$$

$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l, \quad l = 1, \ldots, L \quad (3)$$

$$y = \text{LN}(z_L^0) \quad (4)$$

- The Vision Transformer (ViT) encoder consists of alternating layers of multiheaded self-attention and MLP blocks. Layernorm (LN) is applied before every block, and residual connections placed after every block. The MLP contains two layers with a GELU non-linearity.

04. Experiments & Results

Optimized Models	Train Loss	Train R^2	Valid Loss	Valid R^2	Test Score
Simple CNN in 10 (12 epochs)	1.1592	−0.0186	1.0570	−0.0126	N/A
Original VGG16 (12 epochs)	1.2751	0.0112	1.2649	0.0131	N/A
VGG16 w/ Pre-trained weights (12 epochs)	0.9360	0.0269	0.9253	0.04956	N/A
Resnet50 (11 epochs)	0.9326	0.0210	0.9220	0.0481	N/A
ViT (5 epochs)	0.2695	0.4030	0.2919	0.3861	0.286
SWIN Transformer (6 epochs)	0.0780	0.8386	N/A	N/A	N/A
SWIN Transformer (0.3 train, 6 epochs)	0.0579	0.8817	N/A	0.1944	0.26756
Final Ensemble Model	N/A	N/A	N/A	N/A	0.346

RESNET-50 & VGG-16

- Image input → Main Task → Head
- Tabular input → Auxiliary Task → Aux Head

*We assign more weight to the Head than the Aux Head since Head is our main task, and our evaluation metric is calculated for the Head, not the Aux Head.

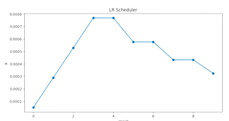


Figure 12. Learning Rate Scheduler

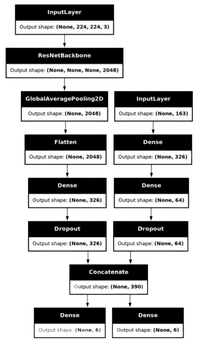
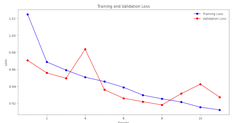
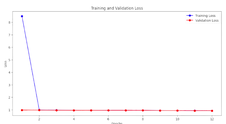


Figure 11. Architecture with Resnet50 Backbone



ResNet-50: Training (blue) vs. Validation (red) Loss



VGG-16: Training (blue) vs. Validation (red) Loss

Best Val R^2_{log}	Best Epoch	Best Val loss	# Parameters (Trainable/Total)
0.0481	7	0.9220	24, 255, 090/24, 308, 210

Table 2. ResNet-50 Training Results

Best Val R^2_{log}	Best Epoch	Best Val loss	# Parameters (Trainable/Total)
0.0496	11	0.925	119, 884, 004/134, 598, 692

Table 3. VGG-16 Training Results

After extensive hyperparameter tuning techniques, including adjustments to layers, dropouts, and freezing/unfreezing, to ResNet-50 and VGG-16, we made a strategic shift to adopt transformer architectures.

SWIN TRANSFORMER

We fine-tune three different pre-trained Swin Transformer models that take in plant images as input and output 6 plant trait values:

1. SwinV2 Tiny Window Size 16x16, Image Size 256 x 256: 28.3M parameters
2. SwinV2 Small Window Size 16x16, Image Size 256x2563: 49.7M parameters
3. Swin Large Window Size 12x12, Image Size 384x384: 196.7M parameters

Swin Transformer Model	Train R^2_{log}	Train R^2_{orig}	Validation R^2_{log}	Validation R^2_{orig}	Test R^2_{orig}
SwinV2 Tiny baseline	0.5726	X	X	0.0402	**0.26691**
SwinV2 Tiny second (12 epochs)	0.7448	X	X	0.0367	0.21284
SwinV2 Tiny second (8 epochs)	0.6317	X	X	0.03954	0.24936
SwinV2 Tiny log-adjusted	X	0.0795	X	0.0188	0.12825
SwinV2 Small	0.6593	X	0.364	X	**0.27308**
Swin Large	0.8812	X	X	0.053	**0.27745**

Table 4. Summary of Swin Transformer Models Performance

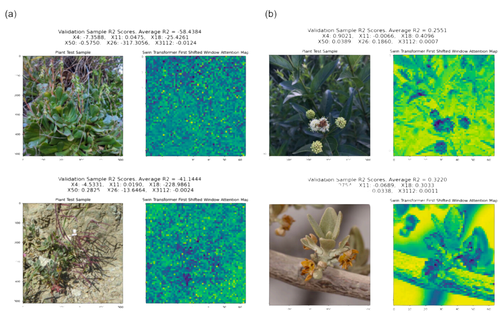


Figure 18. (a) SwinV2 Tiny baseline (a) Bad Valid Attention Maps (b) Good Valid Attention Maps

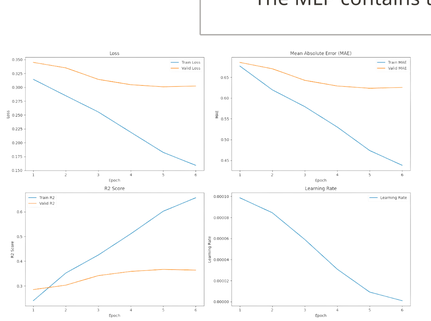


Figure 17. SwinV2 Small Model Training vs. Validation Statistics of Loss, R^2_{log}, MAE, and Learning Rate

VIT

We were able to achieve high R2 scores during training (0.8-0.9) but the model demonstrates significant overfitting. Hence, several key strategies were implemented, including:

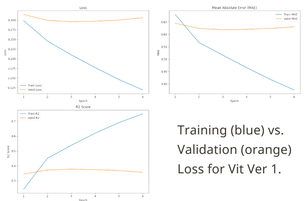
- Weight Decay Application: Utilizing the AdamW optimizer with a weight decay parameter set to 0.01 ensures effective L2 regularization.
- Dropout Regularization (p=0.5) within ViT model architecture
- Early Stopping Mechanism based on validation loss monitoring.
- Batch Normalization to layer inputs.

Best Val R2	Best Epoch	Best Val loss
0.3258	5	0.3159

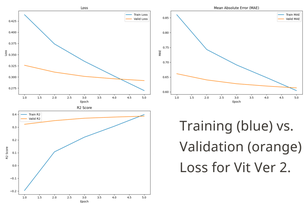
Table 3. Training Results

ENSEMBLE MODEL

We applied ensemble modeling to combine the Swin Large Transformer (test R2 = 0.27745) and ViTForRegression (test R2 = 0.28584) by averaging their predictions, achieving a final test R2 score of **0.34612.** This indicated that the ensemble successfully harnessed the complementary feature representations and predictive strengths of both models.



Training (blue) vs. Validation (orange) Loss for Vit Ver 1.



Training (blue) vs. Validation (orange) Loss for Vit Ver 2.

06. Conclusion

We explored various algorithms for predicting plant traits from images, finding that traditional CNN models like VGG-16 and ResNet-50 struggled, while transformer models like ViT and Swin Transformer performed well due to their ability to capture long-range dependencies. To enhance performance, further investigation into parameter-efficient fine-tuning techniques like LoRA and DoRA, as well as regularization methods like the SMART algorithm, is recommended.

07. Selected References

Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn et al. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

C. Schiller et al. Deep learning and citizen science enable automated plant trait predictions from photographs. Scientific Reports, 11(1):16395, Aug. 2021. Publisher: Nature Publishing Group.