Understanding Plant Traits with Vision Foundation Models

Shu-wen Yang（楊書文）[1] Chiou-Shann Fuh（傅楸善）[2]
[1]Graduate Institute of Communication Engineering
[2]Department of Computer Science and Information Engineering
[1]f08944041@ntu.edu.tw [2]fuh@csie.ntu.edu.tw

Abstract

Analyzing plant traits from images is important, as the information is essential for understanding the ongoing environmental change of earth. However, the vast amount of various species of plants marks the difficulty of this task, since annotating all the existing plants with detailed properties is time and cost-prohibitive, thus naively scaling-up labeled data is not a feasible solution. On the other hand, the foundation model paradigm facilitates knowledge transfer from powerful pre-trained models to various downstream tasks. These models are particularly advantageous for adapting to new data domains with limited downstream computational and labeling resources, where the general knowledge of the foundation models are acquired from web-scale data in advanced. In this study, we then propose to employ vision foundation models to mitigate the data scarcity issue of plant trait analyses. We explore the limits with different ways of leveraging these models, including different layers for representation extraction, different representation aggregation methods, and different downstream modeling approaches. The techniques are verified by the plant traits prediction challenge, ***PlantTraits2024-FGVC11****, hosted on Kaggle*[1]*, and the software is open-sourced*[2]*.*

1. Introduction

The profound impact of plant traits on ecosystem dynamics and human well-being cannot be overstated. Plant traits, encompassing various properties such as leaf area, plant height, and canopy structure, serve as crucial indicators of how plants interact with their environment and compete within ecosystems. As the planet faces accelerating changes due to climate change, understanding these traits becomes essential. Analyzing plant traits on a global scale allows us to predict and mitigate the impacts of environmental shifts on biodiversity, ecosystem productivity, and services. Specifically, this understanding would help ensure food security, protect biodiversity, and sustain human life on Earth amidst the evolving climate scenario.

Despite being critical, plant trait analysis remains an unsolved problem. The difficulty mainly lies in the data scarcity issue: the inaccessibility of high-quality and large-scale labeled data. Supervised learning has long been an simple and effective paradigm to solve problem. In the era of deep learning, the model can further benefit from scaling law [15]. By simply scaling up the model size and the labeled data amount, the model performance improves consistently. However, this direction is infeasible for plant trait analysis, as annotating plant images with specific properties requires domain expertise. Furthermore, the vast amount of plant species and their diverse coverage aggravate the situation.

The foundation model paradigm [3] could be helpful for resolving the above problems. The paradigm proposes to pre-train a centralized and gigantic model with web-scale data in advanced, and then transfer the acquired general knowledge to various downstream tasks. Since most of the knowledge are learned in the first stage, the second stage only requires simple fine-tuning with limited downstream labeled data. This paradigm is thus especially suitable for the low-resource downstream senarios such as plant trait analysis. Given the success of this paradigm in various data domains including NLP [17, 23], CV [1, 26], and speech [5, 19, 34], we propose to leverage this paradigm for plant trait analysis in this work. Our findings are summarized in the following:

- We present the first study on leveraging vision foundation models for plant trait analysis. We employ two types of state-of-the-art (SOTA) vision foundation models, namely Self-Supervised Learning (SSL) and Vision-Language multi-modality learning. We verify that the SSL approach is more competitive on this task, potentially due to the more general knowledge acquired from larger amount of unlabeled data.
- We explore the hidden layers of representations of the

[1] https://www.kaggle.com/competitions/planttraits2024

[2] https://github.com/leo19941227/plant_traits_2024

SSL model, specifically DINOv2 [22]. We find that the deeper layers are more preferable for plant trait analysis since the task requires high-level semantic information.

- We experiment two types of downstream modeling approaches: the classic Boosting method and the neural network. We find that using deep layers of neural network is essential for surpassing the classic XGBoost [6], and using more layers provide a stable approach to improve performances.
- We find that the labels of the training data of the ***PlantTraits2024-FGVC11*** challenge is very noisy. Careful data curation is necessary for all the methods to function normally. Specifically, we propose to discard the training data points with outlier labels.

2. Plant Trait

Plant functional traits offer valuable insights into plant ecology and functional diversity. Leaf traits such as **Leaf Area (LA)** and **Specific Leaf Area (SLA)** influence photosynthetic capacity and resource acquisition. Tropical rainforest species like banana have large LA, while desert cacti minimize LA to reduce water loss. SLA, the ratio of leaf area to dry mass, is high in fast-growing dandelions and low in drought-tolerant aloe. **Leaf Nitrogen Concentration (LNC)** indicates photosynthetic efficiency, with high LNC in soybean and low LNC in desert succulents. Stem traits, particularly **Stem Specific Density (SSD)**, reflect mechanical support and growth strategies. Hardwood teak has high SSD, providing structural resilience, while softwood willow has low SSD, enabling rapid growth. **Seed Mass (SM)** affects dispersal and establishment success. Coconut produces large seeds that ensure seedling survival, while dandelion generates lightweight seeds for dispersal. Lastly, **Growth Height (GH)** indicates competition for light, with redwood towering over shorter shrubs like rosemary. In this work, the six plant traits mentioned above are predicted given an input image.

3. Related works

In this work, our major focus is on leveraging vision foundation models for plant trait analysis. The related works include different types of vision foundation models and previous works on plant trait analysis.

3.1. Vision Foundation Models

Designing vision foundation models involve deciding the pre-training data type and the learning objectives. Data scale is extremely important. There are majorly two major types of large-scale pre-training data: Unlabeled images and weakly-labeled images. The former are naturally presented on internet and is easy to scale, and the latter relies on the vast amount of images in internet which includes the *alt description*. As a result, the weak paired relationship between vision and text is also naturally presented on the internet and the data is easy to scale. Learning from the first type of data is called Self-Supervised Learning (SSL) and the second type of data is Vision-Language representation learning.

3.1.1 Self-Supervised Learning

The followings are some representative works on vision SSL. CPC [21] and SimCLR [7] present the contrastive learning frameworks that learn to distinguish between different images or different augmented versions of the same image by maximizing agreement between differently augmented views of the same data. MoCo (Momentum Contrast) [12] extends this concept by maintaining a dynamic dictionary of encoded representations to provide a consistent set of negative samples for contrastive learning. Moreover, non-contrastive approaches like BYOL [11] and DINO [4] remove the reliance on the negative samples by simply maximizing the agreement between the positive pairs. Removing the normalizing term results in representation collapsing issue, where the model can trivially outputs constant representation to minimize loss. BYOL and DINO find it important to maintain a slowly changing momentum teacher network to avoid the collapse issue. In this work, we leverage the SOTA vision SSL model DINOv2 [22], which employs the same network architecture as DINO and is pre-trained on a much larger amount of unlabeled data.

3.1.2 Vision-Language Representation Learning

Vision-language representation learning has significantly advanced through the development of integrated models that effectively fuse and leverage both visual and textual information. Models like ViLBERT [20] and LXMERT [29] exemplify the dual-stream architecture approach, where separate pathways for image and text are interconnected through co-attentional transformer layers, enhancing mutual context understanding for tasks like visual question answering. CLIP [24], takes a different approach by using a contrastive learning framework to align images and texts in a joint embedding space, facilitating robust transfer learning and zero-shot capabilities. Similarly, ALIGN [14] scales this concept by training on a larger corpus of data, further refining the alignment between visual and textual representations. These models underscore the potential of vision-language representation learning to develop more nuanced and context-aware systems that perform a wide range of multimodal tasks. In this work, we study the effectiveness of vision-langauge representation learning with CLIP which is open-sourced.

3.2. Plant Trait Analysis

Plant trait analysis is pivotal for understanding ecological interactions and evolutionary patterns within plant communities. The emergence of global databases such as TRY has been instrumental in aggregating and standardizing plant trait data, which has been extensively used in ecological modeling and comparative studies [16]. Advanced statistical and machine learning methods, including generalized linear models and deep learning, are increasingly employed to analyze these data, revealing insights into trait-environment relationships [27]. Moreover, high-throughput phenotyping and remote sensing technologies have revolutionized the way traits are measured, enabling more scalable and detailed assessments [9]. Research often focuses on key functional traits such as leaf area, plant height, and photosynthetic capacity, which are crucial for understanding plant performance and survival under varying environmental conditions [33]. The integration of genomic data with phenotypic traits is also burgeoning, offering deeper insights into the genetic underpinnings of trait variation and adaptation [10]. This multidisciplinary approach enhances our predictive understanding of plant responses to environmental changes and informs conservation and management strategies. In this work, we go beyond classical statistical machine learning methods and leverage the latest vision foundation model techniques to analyze the plant traits.

4. Methods

4.1. Leveraging Vision Foundation Models

We explore both vision SSL and vision-language representation learning for plant trait analysis. Specifically, we pick DINOv2 [22] for the former and CLIP [24] for the latter. Both models rely on the ViT architecture [8]. The architecture is composed of multiple layers of Transformer encoder block [31]. The model takes an image as input, tokenize it into several patches as a 1-D sequence, and then feed the sequence to the Transformer encoder block. The patches are then encoded by several layers of self-attention mechanism. Finally, the temporal representations are pooled into a single vector embedding the image. In this work, we explore leveraging the final pooled embedding for plant trait prediction along with the multiple layers of hidden states we obtained when propagating through several Transformer encoder blocks. To aggregate the temporal dimension of the hidden states, we explore mean pooling and statistical pooling, which concatenates the mean vector and the standard deviation vector.

4.2. Downstream Models

After the steps in Section 4.1, we can represent each image with a single embedding, either the original official pooled embedding or the pooled embedding obtained by aggregating across time axis. To predict the plant trait properties given this embedding, we explore two options: XGBoost [6] and multi-layer perceptrons (MLP). For MLP, we further explore different number of layers and hidden sizes. We use GeLU [13] for the non-linearity in MLP. This comparison aims to understand the necessity of using deep learning models for plant trait analysis and whether the newly collected training data in the challenge is sufficient for fitting deep learning networks. For each plant trait, we train a separate downstream model. When training the downstream MLP model, we use Mean Squared Error (MSE) as the learning objective. We did not find significant boost from multi-task learning. In fact, multi-task learning is frequently worse.

4.3. Data Curation

4.3.1 Training Data Creation

The newly released training dataset in this challenge relies on the species identification apps (examples are iNaturalist or Pl@ntNet). These apps help the users recognize unknown plants in the wild for their species. On the other hand, iNaturalist database includes plant trait data that scientists have been curating for decades for various species. Combining both information, the challenge organizers link the images and their potential plant trait properties with iNaturalist database given the recognized species names. Specifically, two different images with the same recognized species name were labeled by the exactly same 6 plant trait properties (See Section 2).

4.3.2 Noisy Label Issue

Given this information, we can infer that the training data is weakly labeled, as the annotation process is automatic without manual check from human experts. On the other hand, to make the evaluation results credible, we assume that the testing set must contains high-quality annotations. As a result, the training data is much more noisy than the testing data. This situation raises two concerns:

- The noisy (incorrect) training annotations could hurt the model performance on the testing data
- The noisy (incorrect) validation annotations would produce misleading validation results and misguide the model development.

Note that in practice, the training set and validation set are disjoint subsets from the original training data given by the challenge organizers.

4.3.3 Outlier Filtering

To mitigate the above issues, we propose to curate data with the following procedure. First, identify the outliers of each plant traits. Specifically, we mark a training/validation sample as invalid if any of its plant trait value fall outside the 1% and 99% interval among all the values of that trait. Intuitively, we consider extreme plant trait values as abnormal and discard the samples with these values.

5. Experiment

5.1. Dataset

We follow the plant trait challenge, **PlantTraits2024-FGVC11**, hosted on Kaggle[1], including the datasets and the evaluation metric. Specifically, the challenge offers a comprehensive plant trait dataset that combines trait observations from the TRY database with plant photographs from the iNaturalist database. The dataset links species-specific mean and standard deviation trait information from the TRY database to citizen science plant photographs based on matching species names. To enrich the dataset, ancillary predictors derived from globally available raster data were also included. These predictors were associated with each plant photograph using geocoordinates and include:

1. **WORLDCLIM**: temperature and precipitation data
2. **SOIL**: global soil grids dataset with various soil properties like sand content and pH value
3. **MODIS**: satellite data measuring optical reflectance across multiple wavelengths
4. **VOD**: radar constellation data sensitive to plant water content and biomass.

These geospatial datasets provide supplementary environmental context to the plant photographs, enabling a more comprehensive analysis of plant traits. These context information serve as ancillary variables to facilitate potential better prediction of the six target plant trait, as illustrated in Section 2. However, in this study we focus on studying the quality of visual representation alone, so we do not include these ancillary variables as the model inputs.

5.2. Training and Evaluation Details

We pre-process the images following the official pipeline of the vision foundation model. For example, for DINOv2 the pre-processing steps include resizing the input image input into 224 × 224 and normalize each RGB channel with the pre-defined mean and variance. We leverage the *Transformers* library [32] for the standard image processing steps and the embedding extraction pipeline. For training the downstream MLP models, we use Adam [18] optimizer with the initial learning rate of 0.0001 without explicit learning rate scheduler. To create our offline standard split for development, we randomly (but deterministically) split the original training data from the organizers into 90% and 10% subsets for training and validation respectively. When reporting the evaluation results, we report the mean R2 score which is an average across the 6 separate R2 scores for individual plant traits (See Section 2). We report the mean R2 score on three evaluation sets: our offline validation set, Kaggle public set and Kaggle private set.

Table 1. The number of data samples of the training and validation sets under each filtering strategy.

Filtering	Training	Validation
No	49940	5549
1% ~ 99%	45233	4993
5% ~ 95%	29836	3274

5.3. Effectiveness of Outlier Filtering

Before directly jump into the modeling approach, we first examine the dataset quality. This examination is an important step to ensure our offline evaluation set is reflecting the Kaggle public score, so that we are guided to develop models which are more robust against domain shift. With this robustness, we avoid risks of overfitting on the Kaggle public set. Furthermore, examining data quality help filter out invalid training samples which hurt model performances.

In the following experiments, we employ DINOv2 as the embedding extractor and use the default pooler output as the image embedding. We train a XGBoost model for each plant trait given the image embeddings. Thus, 6 models are trained. We set the different optimization iterations of XGBoost to simulate the different model performances. This simulation is to examine the correlation among the validation score, Kaggle public score, and Kaggle private score.

Firstly, we do not filter out any samples. Both training and validation sets are noisy. As shown by Table 2, all three scores are meaningless and are not well correlated. This might due to that training and validation samples are too noisy, which leads to nearly random results.

Next, we try to filter out a few extreme values for each plant trait. Specifically, only training samples whose all six plant trait values reside in the 1% ~ 99% are kept. With this filtering, the amount of training and validation data decrease, as shown by Table 1. We also show the training and validation result in Table 2. The results show that filtering out a few extreme values have huge impact on the model training stability, where the validation scores align well with public scores and private scores[3]

Finally, we further experiment with a more aggressive filtering to only use samples with trait values within the range of 5% ~ 95%. This filtering results in halving the amount of both training and validation data as shown by Table 1. The results in Table 2 show that despite the resulting validation set still correlating well with Kaggle public

[3] We collect the private scores after the competition is due with late submissions.

Table 2. We compare the R2 ↑ score of different filtering thresholds for determining the outlier samples. *Valid* means the R2 score on our offline validation set; *Public* and *Private* means the public and the final private score on Kaggle. The experiment is conducted with DINOv2 default image embedding and XGBoost as the downstream model. Different optimization *Iterations* are used to simulate different model performances.

Iterations	No filtering			Use 1% ∼ 99%			Use 5% ∼ 95%		
	Valid	Public	Private	Valid	Public	Private	Valid	Public	Private
10	-2487.63	-53053468.73	-23049345.50	0.17	0.16	0.16	0.16	0.09	0.09
50	-7587.17	-129202866.27	-684826347.35	0.34	0.31	0.31	0.31	0.21	0.22
100	-11442.67	-123827883.95	-1399241312.82	0.38	0.35	0.34	0.34	0.25	0.25
150	-13355.20	-121663196.97	-2189251893.06	0.40	0.36	0.36	0.36	0.27	0.27
200	-13950.59	-113213039.38	-2631452536.32	0.41	0.37	0.37	0.38	0.29	0.29

Table 3. The R2 ↑ score of using MLP as the downstream model show superior performance compared to the classic XGBoost technique. Each score is the mean R2 score over 6 plant traits on the different evaluation sets. The best result according to the Kaggle public score is highlighted in bold. Note that we can only select the best entry according to the public score. The examined foundation model is DINOv2.

Model	Valid	Public	Private
XGBoost (baseline)	0.411	0.373	0.374
MLP with 768 hidden size			
1 layer	0.437	0.381	0.385
3 layer	0.463	0.374	0.375
5 layer	0.463	0.377	0.388
MLP with 1024 hidden size			
1 layer	0.443	0.377	0.380
3 layer	0.472	0.364	0.398
5 layer	**0.479**	**0.394**	**0.405**
10 layer	0.484	0.393	0.415

and private scores, the severe decrease in the training data amount leads to significant model degradation. As a result, we conclude that outlier filtering is critical and can boost model performances significantly when the filtering hyper-parameters is well-tuned.

5.4. Effectiveness of Neural Network

Given the initial success of outlier filtering and the basic XGBoost technique, we next explore the possibility of fitting an neural network (MLP) given the relative large amount of training data (despite being slightly noisy.) The modeling and training details for MLP is described in Section 4.2 and Section 5.2. We explore different number of hidden layers and different hidden sizes in this section since neural networks can rely on the scaling law [15] to improve model performances.

Table 4. Exploring the R2 ↑ score of using hidden states as the image embedding. Mean pooling and statistical pooling are used for collapsing the time axis. The examined foundation model is DINOv2. The *Pooled Output* row uses the default model output, which is already a single embedding.

Model	Mean Pooling	Statistical Pooling
Layer 0	0.037	0.039
Layer 3	0.074	0.083
Layer 6	0.102	0.118
Layer 9	0.139	0.153
Layer 12	0.176	0.198
Layer 15	0.233	0.248
Layer 18	0.289	0.291
Layer 21	0.335	0.348
Layer 24	0.382	0.391
Pooled Output	0.479	

As shown by Table 3, the benefit of using MLP is not evident when only using the hidden size of 768 with less than 5 hidden layers. Despite the higher validation score, the actual Kaggle public scores do not improve significantly. When increase the number of hidden layers to 5, the model starts to surpasses XGBoost significantly[4].

We next try to further improve model performances with the scaling law. Specifically, we enlarge the hidden size from 768 to 1024 to increase the model size. As shown by Table 3, when using more and more layers, the model performances increase significantly and achieve the public score of **0.394**. We conclude that using neural network as the downstream model is effective and can benefit from the neural scaling law.

[4] Note that the hyper-parameters of XGBoost is well-tuned with and extensive grid search.

Table 5. Exploring R2 ↑ using hidden states as the image embedding. Statistical pooling is used for collapsing the time axis. The examined foundation model is DINOv2. The *Pooled Output* row uses the default model output, which is already a single embedding. This table shows the Kaggle scores for the layer-wise results.

Model	Valid	Public	Private
Layer 0	0.039	0.017	0.027
Layer 3	0.083	0.043	0.042
Layer 6	0.118	0.074	0.080
Layer 9	0.153	0.104	0.115
Layer 12	0.198	0.135	0.159
Layer 15	0.248	0.200	0.214
Layer 18	0.291	0.221	0.231
Layer 21	0.348	0.267	0.280
Layer 24	0.391	0.294	0.317
Pooled Output	0.479	0.394	0.405

5.5. Exploring the Hidden Layers

As discovered in NLP [30] and speech [34], when using the frozen foundation models, the last layer of representation is not universal for all the tasks, and careful layer selection should be conducted. We examine whether the internal hidden states of foundation models are potentially better embeddings for plant trait analysis.

The hidden states of foundation models are usually in three dimensions: batch size, number of timestamps, and embedding hidden size. We need to first reduce the temporal representations into a single embedding. We try two prevailing methods for the pooling operation: mean pooling and statistical pooling [28]. The latter involves concatenating mean vector and standard deviation vector. The variance information has been proven critical for speaker verification in [28]. To explore all the layers more efficiently and observe the general trend of the information flow across layers, we examine hidden states for every three layers.

Table 4 shows two major findings. Firstly, statistical pooling is always better than mean pooling in the context of plant trait analysis. Second, unfortunately, all the hidden states of DINOv2 fail to produce promising results compared to the original model output. By verifying with Kaggle submissions in Table 5, we further confirm that the internal layers of DINOv2 do not possess useful information for plan trait analysis.

5.6. Exploring Vision-Language Representation

We explore different types of learning paradigms for pretraining vision foundation models. Specifically, we study the difference between and SSL and Vision-Language with DINOv2 and CLIP. The motivation behind this study driven by the success of Whisper [25]. Whisper is an Automatic Speech Recognition (ASR) model trained by large-scale weakly supervised transcriptions on internet. Despite still being a smaller scale compared to purely unsupervised pretraining [2], Whisper achieves similar or even better ASR performances compared to speech SSL models. The results indicate that SSL and weakly supervised learning are both promising approaches to learn powerful representations. As a result, we aim to understand the representation quality of the weakly supervised vision model, CLIP, which exploits the knowledge from pairs of image and alt text on Internet. We show the result of CLIP in Table [24]. Compared to DINOv2, the representation quality of CLIP is not ideal, and it lags behind DINOv2 by over 7 R2 points. We conclude that, despite DINOv2 and CLIP showing similar performances in regular image classification tasks, their transferability to plant images is hugely different, with DINOv2 being much more robust against domain shifts[2].

Table 6. Exploring R2 ↑ with different types of learning paradigms for vision foundation models. Specifically, we compare SSL with Vision-Language representation learning with DINOv2 and CLIP.

Model	Valid	Public	Private
DINOv2 [22]	0.479	0.394	0.405
CLIP [24]	0.362	0.321	0.334

6. Conclusion

We leverage vision foundation models to mitigate the data scarcity issue in plant trait analyses. Our results show that the state-of-the-art (SOTA) self-supervised learning (SSL) model, DINOv2, achieves competitive performance on this task. Combined with our proposed data curation technique, we achieve scores of 0.394 and 0.405 on the Kaggle public and private leaderboards, respectively. As a result, we ranked 41st in the PlantTraits2024-FGVC11 Kaggle challenge. In this study, we further explore different directions for improvement, including the hidden states of the vision foundation model and the Vision-Language foundation model. Our results verify that the hidden states do not possess useful information for plant trait analyses and that SSL techniques are generally more competitive than multi-modality learning in this task. Finally, we make our training script available as open source to aid reproducibility[2].

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023. 1

[2] Lo¨ıc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, et al. Seamlessm4t-massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*, 2023. 6

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv e-prints*, pages arXiv–2108, 2021. 1

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve´ Je´gou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2

[5] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16 (6):1505–1518, 2022. 1

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016. 2, 3

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[9] Iryna Dronova and Sophie Taddeo. Remote sensing of phenology: Towards the comprehensive indicators of plant community dynamics from species to regional scales. *Journal of Ecology*, 110(7):1460–1484, 2022. 3

[10] Robert T Furbank and Mark Tester. Phenomics–technologies to relieve the phenotyping bottleneck. *Trends in plant science*, 16(12):635–644, 2011. 3

[11] Jean-Bastien Grill, Florian Strub, Florent Altche´, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[13] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1, 5

[16] Jens Kattge, Gerhard Bo¨nisch, Sandra D´ıaz, Sandra Lavorel, Iain Colin Prentice, Paul Leadley, Susanne Tautenhahn, Gijsbert DA Werner, Tuomas Aakala, Mehdi Abedi, et al. Try plant trait database–enhanced coverage and open access. *Global change biology*, 26(1):119–188, 2020. 3

[17] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 1

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[19] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36, 2024. 1

[20] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 2

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[22] Maxime Oquab, Timothe´e Darcet, The´o Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023. 2, 3, 6

[23] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 6

[25] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023. 6

[26] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever.

Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 1

[27] Franziska Schrodt, Jens Kattge, Hanhuai Shan, Farideh Fazayeli, Julia Joswig, Arindam Banerjee, Markus Reichstein, Gerhard Bo¨nisch, Sandra D´ıaz, John Dickie, et al. Bhpmf–a hierarchical b ayesian approach to gap-filling and trait prediction for macroecology and functional biogeography. *Global Ecology and Biogeography*, 24(12):1510–1521, 2015. 3

[28] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5329–5333. IEEE, 2018. 6

[29] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, 2019. 2

[30] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, 2019. 6

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Re´mi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 4

[33] Ian J Wright, Peter B Reich, Mark Westoby, David D Ackerly, Zdravko Baruch, Frans Bongers, Jeannine Cavender-Bares, Terry Chapin, Johannes HC Cornelissen, Matthias Diemer, et al. The worldwide leaf economics spectrum. *Nature*, 428(6985):821–827, 2004. 3

[34] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. Superb: Speech processing universal performance benchmark. *Interspeech 2021*, 2021. 1, 6