

Machine Learning

Nature's Quest: Deep Learning Exploration of Global Plant Traits from Images and Geodata

Kai E. Niermann Dávid Miklo Trix Taicet Red Kaláb
Conner Dassen

March 29, 2024

Abstract

1 Introduction

Global warming and more broadly climate change have become a major concern for the world as its effects are becoming more and more apparent [12]. Changing weather patterns, especially towards more extreme conditions are causing plants to adapt to new environments [3]. One method of measuring said adaptation is to look at the traits, that is, properties of a plant that describe how it functions and interacts with the environment. These traits include but are not limited to the plant height, leaf area, but also dry mass, and leaf nitrogen content amongst various others. Monitoring these traits allows us to gain vital insights into how climate change impacts different ecosystems. While somewhat simple manual measurement techniques exist, at scale they are not feasible. This is where Convolutional Neural Networks (CNN) come in.

Through the work demonstrated by Schiller et al [10] we know that CNNs can be used to predict plant traits from images. The images used to train this network came from *citizen science photographs* which are images taken by citizens of plants from all across the world using AI plant species identification apps (e.g. iNaturalist, Pl@ntNet). Citizen science photographs also come with location metadata, which can be used to extract ancillary geodata such as precipitation, temperature, and soil type. This geodata can optionally be combined with the images to create a CNN which can potentially learn to extract features from images in conjunction with geodata to predict plant traits.

For our method, we wanted to compare the accuracy of a CNN trained on images alone and using a pre-trained backbone (e.g. ResNet, EfficientNet, etc.) to a CNN trained on images and geodata. We hypothesize that the CNN trained on images and geodata will outperform the CNN trained on images alone. Although the geodata could potentially lead the model to learning features not necessarily helpful for accurate prediction of plant traits due to a multitude of reasons.

2 Method

2.1 Data processing

Integral to any machine learning model is the data used to train, validate, and ultimately test the model. The data used consisted of 3 main components: ancillary geodata from various sources, the main task trait means and auxiliary task standard deviations, and the training images of the plants.

Upon visual inspection one of the first issues we spotted was a considerable chunk (29.53%) of data missing for the auxiliary task standard deviations. Through some previous work on the same Kaggle, we noticed that the auxiliary data was a useful inclusion. We can also validate these reports with previous work applying auxiliary tasks in CNNs such as the work of Lukas Libel and Marco Kroener which confirmed that the inclusion of an auxiliary task with minor relevance to the main task did indeed boost performance [7] or Chen et. al. [2] which demonstrated that for very similar or mostly similar auxiliary tasks the contribution is generally a positive one. Based on this we decided to use a simple k-nearest neighbor (kNN) imputation strategy which has generally proved effective [5]. The rest of the data had all values present.

The geodata was another point of consideration in the data pre-processing phase. Looking at the individual columns we noticed that the instances had features that encoded similar information and thus were likely redundant in the training process. Since we are working with a very high dimensional feature space to get some sort of visual verification of this claim we plotted a correlation heatmap for the 3 biggest groups of the geodata, namely the datasets: SOIL, MODIS, and VOD.

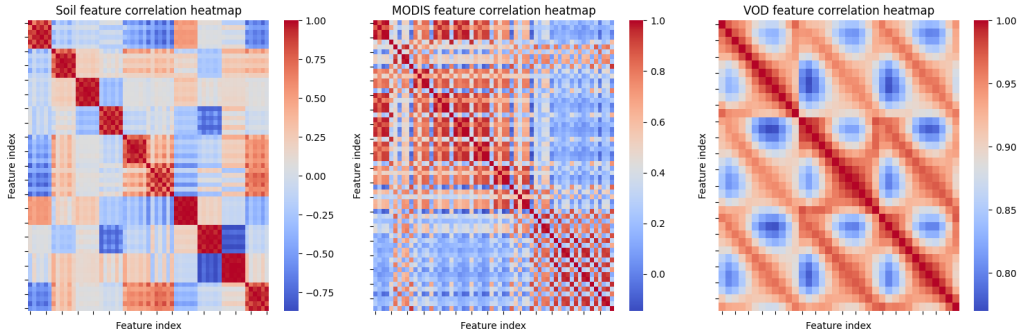


Figure 1: Correlation heatmap of the 3 biggest groups of geodata

Since these datasets individually can be broken down into groups of features generally talking about a similar topic we can see that the correlation within these groups is quite high. Even though previous work [6] has demonstrated the viability of neural network models to learn from high dimensional data, removing redundant features has in numerous instances been shown to improve model performance [1]. With this in mind, we decided to apply a Principal Component Analysis (PCA) to the geodata to reduce the dimensionality of the data. We specified that the number of principal components should be such that 95% of the variance is explained.

2.2 Data preparation

Our training and test image instance - consisting of science citizen photographs - were taken by a wide variety of people under a wide variety of conditions. This means that the images are of varying quality, resolution, and orientation. We can see this just by taking a random sample of 5 images.

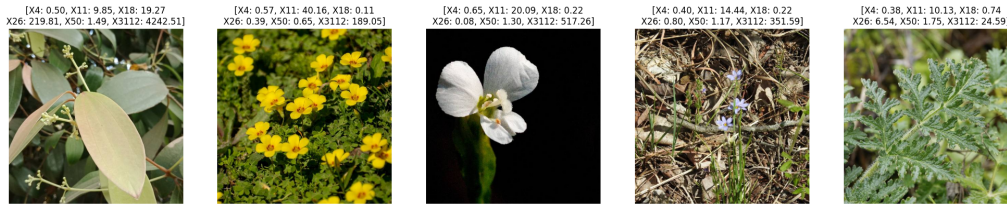


Figure 2: Random sample of 5 images before augmentation annotated with their target trait labels

To prevent our model from fitting to noise in the data as a byproduct of how images were taken we decided to apply different keras preprocessing layers to batches concurrently. A key benefit of this would be that the model can generalize better to unseen data by learning potentially more meaningful features. While this approach of applying augmentations in the preprocessing stage is less flexible than having filters integrated into the model it does reduce model complexity though at the cost of potentially losing some ability to learn more complex features by being able to dynamically adjust the filters. Sampling again we can look at the results of our augmentations.



Figure 3: Random sample of 5 images with augmentations: brightness, rotation, contrast, zoom, hue, cutout, flip, blur, saturation

In addition to augmentation we noted that - though not too common - there was still a presence of certain substantial outliers in our data. Through visual inspection a common reason seems to have been specifically how close an individual was when they took an image. As the training data itself was labelled by a machine learning model in a supervised fashion its evident perspective was not something the initial model had learned. As this scaling issue is a byproduct of the initial model which labelled the data and not any true property of the data we decided to remove these outliers. We did this by constraining the target distributions to the interval $[0.0001, 0.9999]$ and then removing any instances that were outside of this interval.

Another method of reducing overfitting we thought relevant to apply here would be k-fold cross-validation. For the choice of k we chose 5 as for large datasets this has proven effective [8] and is generally around the commonly chosen value of 10 [9]. Given that

the data is from a geographically large enough sample we decided to create the folds in a stratified manner under the assumption the distribution of traits would be reasonably representative of the true distribution and thus one the model should learn. This is important as it ensures that the model is trained on a representative sample of the data in each fold and in turn, has its predictions distributed in a similar manner to the assumed true distribution of the traits. So we reduce the risk of overfitting to the training data while also ensuring the model is trained using a representative sample of targets in each fold. To validate that the distributions of the target traits are indeed similar we averaged the distributions of the target traits in each fold overlaid this on the original distribution of trait means and normalized them to the same scale.

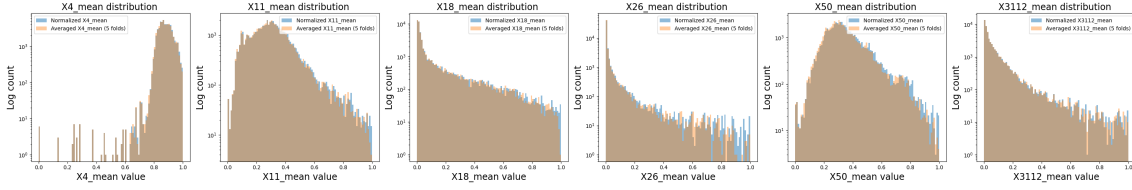


Figure 4: Comparison of the average trait distributions in each fold with the original distribution of trait means ; (blue) normalized distribution of the trait means across the entire dataset ; (yellow) average normalized distribution of the trait means in each fold

2.3 Model design

2.3.1 CNN trained on images

Our image CNN is a multi-output model that takes in a tensor consisting of red, green, and blue channels with an image size of 224x224 pixels. The model consists of a trained backbone; in our instance, we compared *efficientnetv2_b2_imagenet*, *efficientnetv2_s_imagenet*, and *resnet50_imagenet*. The backbone outputs then feed into a global average pooling layer, followed by 2 dense layers and a dropout layer connected to the two dense output layers for the main and auxiliary tasks. The hidden layers use a ReLU activation function, the outputs use no activation function as the targets are continuous. The model is trained using the Adam optimizer with a learning rate of 1e-4, a batch size of 48, and for 12 epochs. Additionally, we used the R^2 metric for both loss and evaluation of the model performance as this is a regression problem. Finally, we used step-based for the learning rate schedule which was defined as follows

$$\text{lr} = \text{lr}_{\max} \times d^{\lfloor \frac{e_d}{r} \rfloor}$$

Where e_d is the current epoch, r is the step size ($=2$), and d is the decay rate.

Residual Network (ResNet) [4] introduced in 2016 is a deep learning model that is able to learn features from images. Most famously it's known for mitigating the vanishing gradient problem through the introduction of identity skip connections. We are using ResNet50 which was trained on imagenet and consists of 48 convolution layers, in addition to a MaxPool and AveragePool layer. EfficientNetV2 [11] is a family of CNNs from 2021 that have faster speeds than many previous models primarily by increasing image size during training and incorporating adaptive regularization to compensate for any

potential accuracy loss. We are using *efficientnetv2_b2_imagenet* and *efficientnetv2_s_imagenet* which are trained on imagenet, with the latter being a smaller (fewer parameters) version of the former. Comparing these backbones in a table we can get some overview of how they stack up as seen in 1

To compare these backbones we trained 3 different models with the same architecture but different backbones. We also specified a contribution of 0.3 to the final loss of the auxiliary task (as opposed to 1.0 for the main task) as to limit its impacts on the direction of training. A visual overview of the different models can be seen in 5.

	EfficientNetV2 b2	EfficientNetV2 s	ResNet50
parameters	8.77M	20.33M	23.56M
imagenet top 5 accuracy	94.9%	96.7%	(missing)

Table 1: Comparison of the different backbones and published top 5 accuracy on imagenet for the specific models

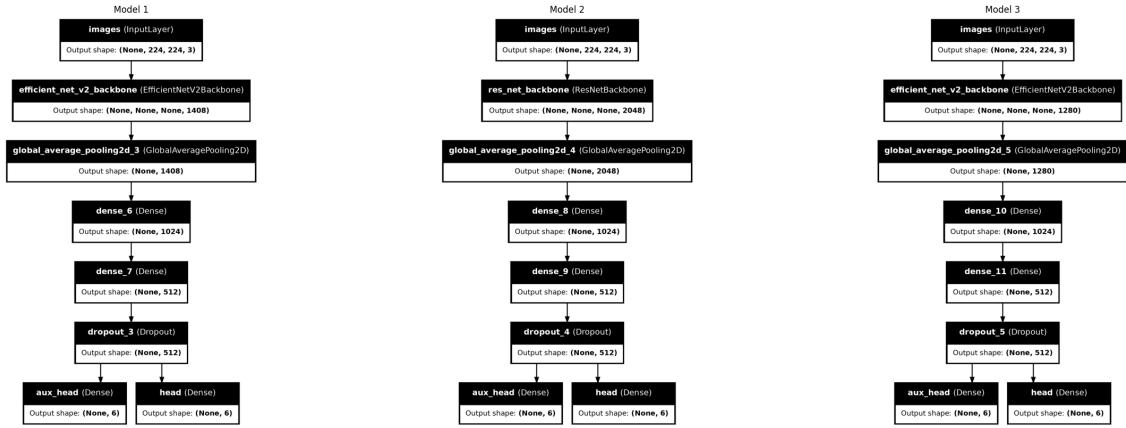


Figure 5: Visual overview of the 3 models with different backbones

2.3.2 Image CNN tuning

Looking at the results of the initial choice of hyperparameters for the different models the first thing we noticed was the similar behavior of all the R^2 values. The coefficient of determination is defined as follows

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

Where SS_{res} is the residual sum of squares and SS_{tot} is the total sum of squares. One observation in our results was that for all instances we had a negative R^2 , implying that $SS_{res} > SS_{tot}$ meaning our model preforms worse than the baseline at $SS_{res} = SS_{tot}$. This is a clear indication that the model is evidently learning nothing from the data, numerous reasons could be the cause of this.

One of the first things we decided to test was the learning rate schedule. We tested out 3 different schedules: step-based (baseline), exponential, and cosine annealing. Due to the results of the initial training not being conclusive in regards to which model is better we decided to simply continue with the model using the *efficientnetv2_b2_imagenet* backbone as it has a high top 5 accuracy while also having fewer parameters, reason being that the source of the problem is likely not the backbone rather but rather to do with the model architecture surrounding this and the hyperparameter choices.

2.3.3 CNN trained on images and geodata

2.4 Model evaluation

3 Results

4 Discussion

5 Conclusion

References

- [1] J.-A. Chen, W. Niu, B. Ren, Y. Wang, and X. Shen, *Survey: Exploiting data redundancy for optimization of deep learning*, 2022. arXiv: 2208.13363 [cs.LG].
- [2] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, “GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks,” in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, Oct. 2018, pp. 794–803. [Online]. Available: <https://proceedings.mlr.press/v80/chen18a.html>.
- [3] S. B. Gray and S. M. Brady, “Plant developmental responses to climate change,” *Developmental Biology*, vol. 419, no. 1, pp. 64–77, 2016, Plant Development, ISSN: 0012-1606. DOI: <https://doi.org/10.1016/j.ydbio.2016.07.023>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0012160616302640>.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, *Identity mappings in deep residual networks*, 2016. arXiv: 1603.05027 [cs.CV].
- [5] L. O. Joel, W. Doorsamy, and B. S. Paul, *On the performance of imputation techniques for missing values on healthcare datasets*, 2024. arXiv: 2403.14687 [cs.LG].
- [6] I. Kansizoglou, L. Bampis, and A. Gasteratos, “Deep feature space: A geometrical perspective,” *CoRR*, vol. abs/2007.00062, 2020. arXiv: 2007.00062. [Online]. Available: <https://arxiv.org/abs/2007.00062>.
- [7] L. Liebel and M. Körner, “Auxiliary tasks in multi-task learning,” *CoRR*, vol. abs/1805.06334, 2018. arXiv: 1805.06334. [Online]. Available: <http://arxiv.org/abs/1805.06334>.

- [8] B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete bayesian network analysis?" *Computational Statistics*, vol. 36, no. 3, pp. 2009–2031, 2021, issn: 1613-9658. doi: 10 . 1007 / s00180 - 020 - 00999 - 9. [Online]. Available: <https://doi.org/10.1007/s00180-020-00999-9>.
- [9] I. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of machine learning algorithms with different k values in k-fold cross-validation," *International Journal of Information Technology and Computer Science*, vol. 6, pp. 61–71, Dec. 2021. doi: 10.5815/ijitcs.2021.06.05.
- [10] C. Schiller, S. Schmidtlein, C. Boonman, *et al.*, "Deep learning and citizen science enable automated plant trait predictions from photographs," *Scientific Reports*, vol. 11, no. 1, p. 16 395, 2021. doi: 10 . 1038 / s41598 - 021 - 95616 - 0. [Online]. Available: <https://doi.org/10.1038/s41598-021-95616-0>.
- [11] M. Tan and Q. V. Le, *Efficientnetv2: Smaller models and faster training*, 2021. arXiv: 2104.00298 [cs.CV].
- [12] L. Wang, L. Wang, Y. Li, and J. Wang, "A century-long analysis of global warming and earth temperature using a random walk with drift approach," *Decision Analytics Journal*, vol. 7, p. 100 237, 2023, issn: 2772-6622. doi: <https://doi.org/10.1016/j.dajour.2023.100237>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2772662223000772>.