

# Plant Species Identification Using Transfer Learning - PlantCLEF 2020

Nanda H Krishna\*, Rakesh M\*, and Ram Kaushik R\*

Department of Computer Science and Engineering  
SSN College of Engineering, Kalavakkam, Chennai, India  
{nanda17093, rakesh17123, ramkaushik17125}@cse.ssn.edu.in

**Abstract.** Automated prediction of plant species from images is immensely useful to conservationists, especially in the case of data-scarce regions of biodiversity. The PlantCLEF 2020 Challenge provides a platform for the creation of a classifier to identify plant species from a large collection of labelled images. The aim of the challenge is to identify which methods work best on the same data, and hence accelerate research progress in the field. In this paper, we discuss the submissions made by our team to the challenge, based on transfer learning. For our submissions, we trained our models on Cloud TPUs and TPU Pods available on Google Cloud Platform. All our models, which were initially trained on the ImageNet Dataset, were fine-tuned to the PlantCLEF 2020 Dataset using transfer learning. With our ResNet-50 models, we achieved an overall MRR of 0.008 in the testing phase. For specifically chosen classes with fewer training samples, we achieved an MRR of 0.003.

**Keywords:** Species Identification · Transfer Learning · TPU

## 1 Introduction

Automated plant species identification from images is immensely useful in data-scarce regions of biodiversity, to identify and record the flora present. With the advent of Deep Learning and novel model architectures, performance in this task has improved considerably over the years. However, classification with a very large number of classes is still a tough task with considerable scope for improvement.

It is with the objective of building a reliable plant species identification system that the PlantCLEF 2020 Challenge [3] was organised, as part of LifeCLEF 2020 [5]. The challenge provides a platform for the evaluation of different methods on the same dataset, in an effort to identify the best-performing algorithm

---

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

\* These authors contributed equally.

for the task. A large labelled dataset of plant images was provided by the organisers, wherein the images exhibit great inter- and intra-class diversity, mimicking the real world.

Evaluation of the submissions to the challenge was based on the Mean Reciprocal Rank (MRR) metric. Additionally, the MRR for the classification of specific species with fewer training samples was considered as a secondary metric.

In this paper, we will discuss our team’s submissions to the PlantCLEF 2020 Challenge in detail. We will first discuss the methodology we employed to solve the task. next, we will outline the resources used to build our models. Finally, we will discuss the results obtained by our submissions, along with an analysis and a note on future work.

## 2 Methodology

### 2.1 Data Preprocessing

First, we normalised the pixel values to the range  $[0, 1]$ . All the images were then resized to  $224 \times 224 \times 3$ , due to limited computational resources. To do so, we first resize the smaller of the two spatial dimensions ( $H$  or  $W$ ) to 224 pixels, and then extract the center crop of size  $224 \times 224$  ( $H \times W$ ). The disadvantage of this preprocessing step is the possible removal of salient information present in pixels that were discarded. An alternative approach could be the use of multiple  $224 \times 224$  crops from the image extracted with a certain stride, to ensure all details in the original image are present in a subset of the extracted images. However, this would increase the number of images in the dataset and thus the computation time.

We applied augmentations to the training images to improve the generalisation performance of our models. Images were randomly zoomed-in or zoomed-out by up to 20% of the image width, rotated by an angle in the range  $[-45^\circ, 45^\circ]$  and flipped about their vertical axis. The objective of these augmentations is to make the models learn robust features, invariant to scale or rotation.

### 2.2 Models Used

**VGG-16 & VGG-19:** We first experimented with the VGG-16 and VGG-19 architectures [9], pre-trained on the ImageNet Dataset [2,8]. The pre-trained models were used as non-trainable feature extractors, and their output for every image was passed to a shallow ANN of 2 Dense layers (each having 4096 units), followed by a Fully-Connected layer (with *softmax* activation). This method did not perform well on our validation set, and we did not make any submissions based on this method.

**ResNet-50:** Both our final submissions to the challenge were made using the ResNet-50 [4] models. We used the ResNet-50 architecture, again pre-trained on the ImageNet Dataset [2,8], as a trainable feature extractor network. Following the layers of the ResNet, we added 2 Dense layers of 2048 units each, followed by a Fully-Connected layer with *softmax* activation.

All models were trained with *Adam* as the optimiser and *categorical cross-entropy* as the loss function. The number of training epochs was set to 8. As there were training examples for only 998 classes, all our models had an output vector of 998 probabilities.

### 2.3 Prediction

A single species could have one or more images associated with the same submission ID. Two techniques were used to make predictions for the individual species. The predictions were made for the individual images, and either the average or the maximum of the probabilities predicted were used to rank the species and generate the final predictions.

## 3 Source Code and Resources Used

Our models were built using the Keras [1] Deep Learning framework, and the code was written in Python 3. Jupyter Notebooks containing our code have been made publicly available on our GitHub Repository<sup>1</sup>.

All of the work was performed on the Google Cloud Platform (GCP). Tensor Processing Units (TPU) are processors developed by Google specifically for the purpose of accelerating tasks that involve computation on tensors. TPU Pods are a collection of TPUs that are connected by a high speed network. GCP provides access to Cloud TPUs and Cloud TPU Pods. Both individual TPUs (version 2 with 8 cores and version 3 with 8 cores) as well as a TPU Pod (version 3 with 256 cores) were made use of in the training phase.

For all other tasks, including making of predictions, an `n1-highmem` VM instance was used, with 8 CPU cores and 52GB of RAM. The storage drive used was a 150GB SSD, connected to the instance.

Despite the power of the computational resources available, we were unable to experiment with more powerful models or approaches due to limited availability of credits. Furthermore, the TPUs allocated to us were present in regions other than that of our VM instance, causing delays and increased compute time due to network operations. The training of our ResNet model, for instance, took around 2 hours per epoch, highlighting the importance of computational resources in data-intensive tasks.

<sup>1</sup> <https://github.com/nandahkrishna/PlantCLEF2020>

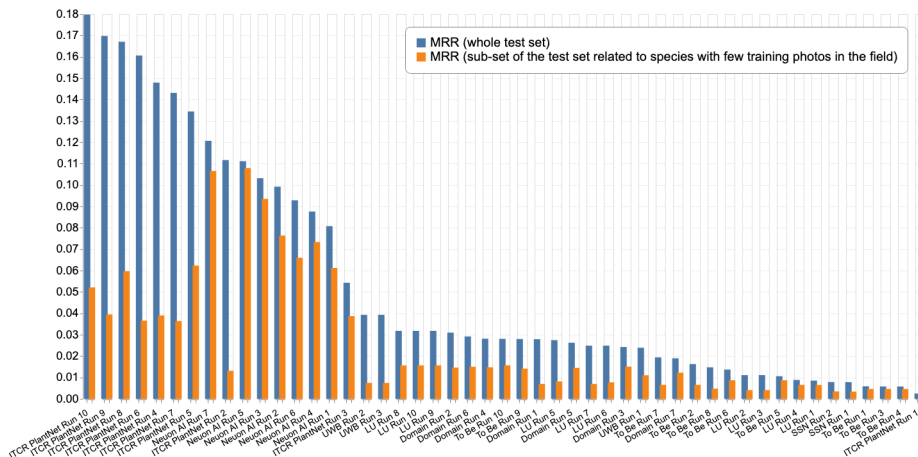
## 4 Results

The results obtained by our models can be seen in Table 1. We have included the *categorical cross-entropy* loss for all our models as well as the MRR score for our submitted models. With both our submissions, we obtained a Testing MRR score of 0.008, with an MRR of 0.003 on the species with fewer training samples.

**Table 1.** Validation and Testing Loss and MRR obtained by our models

Model	Training Loss	Validation Loss	Testing MRR	Testing MRR (Few Species)
VGG-19	2.3316	4.2236	–	–
VGG-16	2.1365	3.0349	–	–
ResNet-50 Avg	1.7417	2.2261	0.008	0.003
ResNet-50 Max	1.7417	2.2261	0.008	0.003

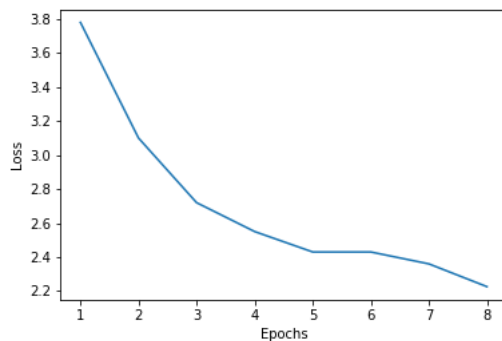
Overall our submissions received the 44th and 45th ranks on the challenge leaderboard. The scores obtained by all submissions to the challenge can be visualised in Fig. 1. The top scores were obtained by teams ITCR PlantNet (Overall MRR 0.180) and Neoun AI (Overall MRR 0.121). On the species with limited training examples, the ITCR PlantNet team obtained a poorer score (MRR 0.062) than the Neoun AI team (MRR 0.108).



**Fig. 1.** Overall results of the PlantCLEF 2020 Challenge

## 5 Conclusion and Future Work

Despite the resources available, training our models was computationally expensive as the large dataset was very large. We were unable to train our models for a larger number of epochs due to the constraints of our cloud computing resources. Increasing the number of training epochs could yield larger gains in classification accuracy or MRR, as indicated by Fig. 2.



**Fig. 2.** Validation cross-entropy loss curve

Additionally, the lower scores obtained by our methods could be attributed to the low similarity between the dataset used for pre-training and the task-specific dataset. In such a setting, using homogeneous domain adaptation techniques such as MMD [6] or approaches based on an architecture criterion [7] could improve performance on the target domain by accounting for the difference in data distribution. We consider only the supervised setting as the PlantCLEF dataset contains sufficient number of examples to facilitate the same. An alternative to this would be the use of heterogeneous domain adaptation techniques using generative models as discussed in [10], which could help improve performance on a target domain which is very different from the source domain, as is in our case.

For the future, we would still consider a transfer learning based approach, however, with pre-training on a similar large dataset. This would considerably improve model performance. In addition, we would like to consider using extreme classification methods to handle the large number of output classes better.

## 6 Acknowledgements

We would like to acknowledge the support received in the form of Cloud TPUs and a Cloud TPU v3 Pod from TensorFlow Research Cloud (TFRC).

## References

1. Chollet, F., et al.: Keras. <https://github.com/fchollet/keras> (2015)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
3. Goëau, H., Bonnet, P., Joly, A.: Overview of the lifeclef 2020 plant identification task. In: CLEF working notes 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778. IEEE Computer Society (2016). <https://doi.org/10.1109/CVPR.2016.90>, <https://doi.org/10.1109/CVPR.2016.90>
5. Joly, A., Deneu, B., Kahl, S., Goëau, H., Ruiz De Castaneda, R., Champ, J., Eggel, I., Cole, E., Bonnet, P., Botella, C., Dorso, A., Glotin, H., Lorieul, T., Servajean, M., Stöter, F.R., Vellinga, W.P., Müller, H.: Lifeclef 2020: Biodiversity identification and prediction challenges. In: Proceedings of CLEF 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)
6. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 2208–2217. PMLR (2017), <http://proceedings.mlr.press/v70/long17a.html>
7. Rozantsev, A., Salzmann, M., Fua, P.: Beyond sharing weights for deep domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(4), 801–814 (2019). <https://doi.org/10.1109/TPAMI.2018.2814042>, <https://doi.org/10.1109/TPAMI.2018.2814042>
8. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* **115**(3), 211–252 (2015), <https://doi.org/10.1007/s11263-015-0816-y>
9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), <http://arxiv.org/abs/1409.1556>
10. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), <https://openreview.net/forum?id=Sk2Im59ex>