A Global-Scale Plant Identification using Deep Learning: NEUON Submission to PlantCLEF 2022

Sophia Chulif[1,2], Sue Han Lee[1] and Yang Loong Chang[2]

[1]Swinburne University of Technology Sarawak Campus, 93350, Sarawak, Malaysia
[2]Department of Artificial Intelligence, NEUON AI, 94300, Sarwak, Malayisa

Abstract

With the increasing knowledge of plants globally, it is becoming difficult for human experts to identify plants manually and systematically. Vascular plants alone are estimated to be more than 300,000 species. However, deep learning methods have recently made progress in automating plant identification. The PlantCLEF 2022 challenge this year aims to tackle the problems faced in global plant identification. With the aggregation of various data from different sources, it is a real problem to deal with big data consisting of many classes, unbalanced classes, inaccuracies, duplications, and a diversity of visual contents and quality. Given a training dataset of 4 million images and 80,000 species, the task of the challenge was to identify the correct plant species from 26,868 multi-image plant observations. This paper describes the submissions made by our team to PlantCLEF 2022. We trained several deep learning models based on the Inception-v4 and Inception-ResNet-v2 architectures. The types of networks constructed were a single convolutional neural network (CNN) and a triplet network. They were either initialised on weights pre-trained from the ImageNet dataset or the weights pre-trained from PlantCLEF 2022 dataset. Although we intended to compare the performance between our single CNN and triplet models, unfortunately, we did not manage to obtain the complete results due to resource and time constraints. Nevertheless, we submitted nine runs and our best submission achieved a Macro Averaged Mean Reciprocal Rank score of 0.6078, placing 4th among the 45 submitted runs. In addition, we have shown that web or noisy data does improve generalisation in the identification. Moreover, the ensemble of models from different network architectures, i.e., Inception-v4 and Inception-ResNet-v2, give higher accuracy than a single model.

Keywords

plant classification, convolutional neural network, deep learning, machine learning, computer vision

1. Introduction

The LifeCLEF Plant Identification Challenge (PlantCLEF) [1] is part of the Conference and Labs of the Evaluation Forum (CLEF), which tackles various multilingual and multimodal information access evaluations [2]. This year, the focus of PlantCLEF 2022 was to classify 80,000 plant species. Compared to its past editions, PlantCLEF 2022 offered the largest ever number of classes for training, making it resource-intensive, which is often the case in real-world applications. In the context of global plant identification, the aggregation of various data from different sources poses many challenges. It is a real problem to deal with big data consisting of many classes, unbalanced classes, inaccuracies, duplications, and a diversity of visual contents and quality. The total training datasets provided in this challenge consisted of 4 million images

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
schulif@swinburne.edu.my (S. Chulif); shlee@swinburne.edu.my (S. H. Lee); yangloong@neuon.ai (Y. L. Chang)

Table 1
Details of the Train, Validation, and Test datasets.

Dataset	No. of images	No. of species
Train (Trusted)	2,821,933	80,000
Train (Trusted + Web)	3,893,560	80,000
Validation	63,119	63,119
Test	55,306	-

and were grouped as "trusted" or "web". In addition, the metadata included more levels of plant taxonomy, i.e., class, order, family, genus, and species. This paper presents our approach, the network architectures used, the training setup implemented, and the results obtained from our submissions to PlantCLEF 2022.

2. Methodology

2.1. Data

Two training datasets were downloaded: "trusted" and "web". After removing some duplicates with the same name, the trusted dataset totalled 2,885,052 images. It is derived from academic sources and collaborative platforms, signifying a higher certainty of quality. Meanwhile, the web dataset totalled 1,071,627 images. It is based on search engine queries and suffered from notable errors, which were then semi-automatically cleaned by the organisers. In addition, from the trusted dataset, we segregated 63,119 images to serve as our validation dataset. This validation dataset consists of unique species belonging to one plant observation id from our trusted train dataset. Lastly, the test set consisted of 55,306 images from 26,868 plant observations. The details of the datasets used are represented in Table 1.

2.2. Network Architecture and Models

The networks implemented in our approach were based on the Inception-v4 and Inception-ResNet-v2 architectures [3]. Two types of networks were constructed: a single convolutional neural network (CNN) and a triplet network. They were either initialised on weights pre-trained from the ImageNet dataset [4] or the weights pre-trained from PlantCLEF 2022 dataset.

Motivation for Triplet Network Convolutional Neural Networks (CNNs) have effectively solved classification tasks in various domains and even perform comparably equal or better than humans. However, a pre-defined number of classes is required before training. If the classification task is assigned with new labels, the network has to be retrained. In addition, CNNs work best when there is sufficient training data.

In a global-scale plant identification task, it is impractical to retrain the network every time a new species is discovered. Furthermore, many plant species, especially those in remote areas, are rarely photographed. More often than not, the available data for training follows a long-tail

Table 2
Details of the multi-task classification labels in the Train dataset.

Label	No. of classes
Class	8
Order	84
Family	483
Genus	9,603
Species	80,000

distribution, resulting in the CNN performing well in classes with many training data and poorer in classes with few or no training data.

Considering these concerns, we opted to implement the metric-learning-based triplet network. Its goal is to learn the similarity and dissimilarities between classes instead of directly classifying them. The network accomplishes this by minimising the embedding distance of the same species while maximising the embedding distance of different species. A small embedding distance indicates the same species. Meanwhile, a large embedding distance indicates different species. Besides, it does not require a pre-defined number of classes before the training. Moreover, as shown in the previous PlantCLEF editions [5, 6], our triplet networks [7, 8] can generalise plant species equally well with or without less training data than a conventional CNN.

Single CNN This network resembles a conventional Inception-v4 and Inception-ResNet-v2 neural network. Similarly, it consists of convolutional layers, pooling layers, dropout layers and fully-connected layers, which return the softmax probabilities of its prediction. The multi-task classification is adopted in this network by utilising the five taxonomy labels: Class, Order, Family, Genus, and Species. Table 2 shows the multi-task classification labels and their number of classes. The network architecture is visualised in Figure 1 (A).

Triplet Network This network resembles the single CNN mentioned above. However, it consists of two streams and instead of using its fully-connected layer for its predictions, it is used to compute the plants' image embedding representation. In addition, a batch normalisation layer is added, followed by L2-normalisation, and finally, a triplet loss layer [1] to train the optimum embedding representation of the plants. Furthermore, instead of its original 1536 features in the fully-connected layer, we reduced its final feature vector to 500. Due to resource limitation, we did not adopt the multi-classification approach in this training. Only the Species taxonomy label is utilised. The network architecture is visualised in Figure 2 (A).

2.3. Training Setup

The networks were set up using Tensorflow 1.12 [9] and TF-Slim [10] library with the hyper-parameters described in Table 3. Random cropping, colour distortion, and horizontal flipping were also applied to the images during training of the networks. The scripts and lists used are available at https://github.com/NeuonAI/plantclef2022_challenge.

[1]The triplet loss is computed using triplet_semihard_loss function provided in Tensorflow 1.12 [9]

Table 3
Details of the Network Training Hyperparameters.

Hyperparameter	Single CNN	Triplet Network
Batch Size	128	128
Input Image Size	$299 \times 299 \times 3$	$299 \times 299 \times 3$
Optimizer	Adam Optimizer [11]	Adam Optimizer [11]
Initial Learning Rate	0.0001	0.0001
End-layers Learning Rate	0.0001	0.00001
Weight Decay	0.00004	0.00004
Learning Dropout rate	0.2	0.2
Loss Function	Softmax Cross Entropy	Triplet Loss

2.4. Inference Procedure

The two main evaluation methods of the models used the Argmax function and embedding dictionary similarity comparison. Argmax is used in the single CNN, while the embedding dictionary similarity comparison is used in the triplet network. The inference procedure for the single CNN and triplet network are visualised in Figures 1 (B) and 2 (B), respectively. The following steps describe the overall inference procedure.

Single CNN

1. Group the images based on the same observation id (if the test set is used).
2. Augment the validation / test image(s) to 10 variations through cropping and flipping. Note that the 10 different variations include the cropping of the top-right, top-left, bottom-right, bottom-left, and centre of the image. Then, these 5 images are horizontally-flipped to obtain a total of 10 image variations.
3. Feed the augmented validation / test images into the network.
4. Obtain the prediction results of the images. Note that since there are 10 variations of each. image, there will be 10 predictions for each image for each label (Class, Order, Family, Genus, Species).
5. Average the 10 prediction probabilities of each image classification.
6. Obtain the Top-1 and Top-5 accuracy (if the validation set is used).
7. Obtain the Top-30 accuracy (if the test set is used).

Triplet Network Before inference, a sample from the training dataset (trusted) is randomly chosen to create a dictionary list. The dictionary list totalled 592,258 images which consist of 80,000 species. A maximum of ten images represent each species in the dictionary.

1. Group the dictionary images based on the same species.
2. Augment the dictionary images to 10 variations through cropping and flipping. Note that the 10 different variations include the cropping of the top-right, top-left, bottom-right, bottom-left, and centre of the image. Then, these 5 images are horizontally-flipped to obtain a total of 10 image variations.

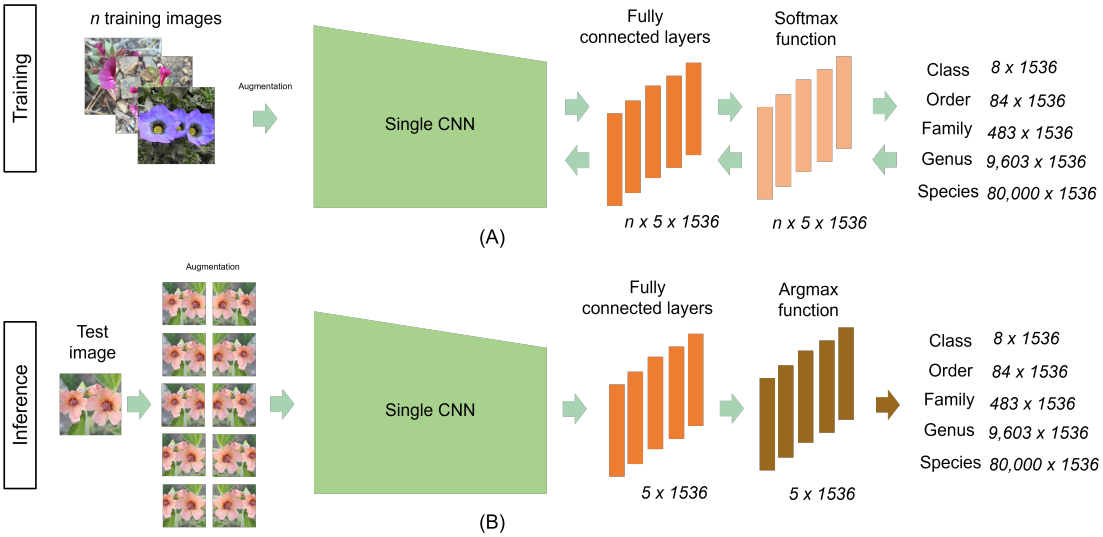

Figure 1: The training and inference procedure of the Single CNN. Figure 1 (A) illustrates the training process of the network. Figure 1 (B) illustrates the inference process of the network.

3. Feed the augmented dictionary images into the network.
4. Obtain the image embeddings of each image. Note that since there are 10 image variations, there will be 10 embeddings for each image.
5. Average the 10 embeddings and save them as a dictionary reference.
6. Repeat steps 2 to 5 until all the 80,000 species embeddings are collected.
7. Group the test image(s) based on the same observation id (if the test set is used).
8. Augment the validation / test image(s) to 10 variations through cropping and flipping as previously.
9. Feed the augmented validation / test images into the network.
10. Obtain the image embeddings of each image. Note that since there are 10 image variations, there will be 10 embeddings for each image.
11. Average the 10 embeddings to obtain the single embedding of each validation / test image.
12. Compute the cosine similarity between the single image embedding and the saved dictionary.
13. Obtain the cosine distance by subtracting the computed cosine similarity from the value of 1.
14. Employ inverse distance weighting on the cosine distance.
15. Acquire the probabilities of the single image embedding mapped to the dictionary.
16. The species mapped to the highest probability denotes the class of the species.
17. Obtain the Top-1 and Top-5 accuracy (if the validation set is used).
18. Obtain the Top-30 accuracy (if the test set is used).
19. Repeat steps 8 to 18 until all the images are evaluated.

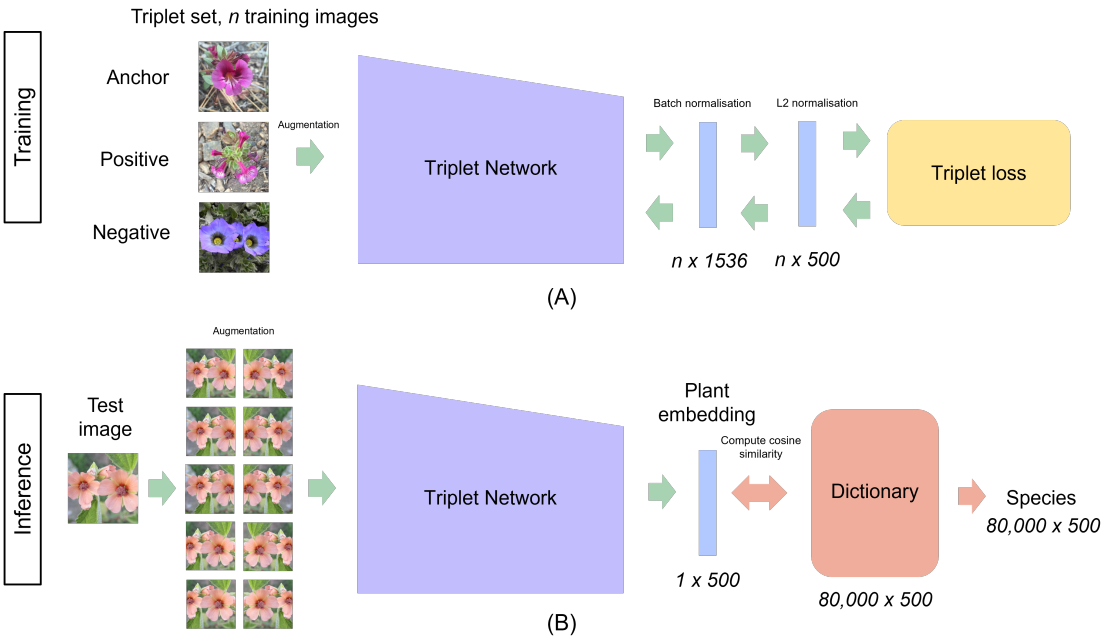

Figure 2: The training and inference procedure of the Triplet Network. Figure 2 (A) illustrates the training process of the network. Figure 2 (B) illustrates the inference process of the network.

3. Experiments

The networks we experimented with were variations of single CNN and triplet networks. They differ in their network architecture, training data, and weights initialised. Table 4 shows the details of our experimented networks. As seen in Table 4, models 1 to 5 are single CNNs, models 6 to 8 are triplet networks, and models 9 to 12 are single CNN whose weights were initialised from the triplet network.

3.1. Results

From the validation dataset, we computed the Top-1 and Top-5 accuracy of the models and tabulated them in Tables 5 and 6, respectively. Due to the large training dataset and time constraints, most of the models trained were not saturated before we evaluated them. In addition, not all the models experimented with were used in the submissions.

Based on our experiments, most of the models trained with a higher number of iterations performed better in the validation dataset. Nevertheless, the higher number of iterations does not necessarily depict higher performance. Comparing Model 2 (421,517 steps) and Model 4 (522,583 steps), Model 2 performed better with a Top-1 accuracy of 0.462 compared to Model 4 of 0.4545. Since the validation set was built from the trusted dataset, which was what Model 2 was trained on, it may have resulted in this bias. Furthermore, we find that the single CNN initialised from the triplet network (Model 12b) performed slightly better than the single CNN initialised from ImageNet (Model 5b). Since our models did not saturate, we cannot give a definite answer

Table 4
Details of the Experimented Models.

Model	Architecture	Train data	Initialised from
1 Single-I (Trusted)	Inception-v4	Trusted	ImageNet
2 Single-IR (Trusted)	Inception-ResNet-v2	Trusted	ImageNet
3 Single-I (Trusted + Web)	Inception-v4	Trusted + Web	Model 1
4 Single-IR (Trusted + Web)	Inception-ResNet-v2	Trusted + Web	Model 2
5 Single-IR (Trusted + Web)	Inception-ResNet-v2	Trusted + Web	ImageNet
6 Triplet-I (Trusted)	Inception-v4	Trusted	ImageNet
7 Triplet-IR (Trusted)	Inception-ResNet-v2	Trusted	ImageNet
8 Triplet-IR (Trusted + Web)	Inception-ResNet-v2	Trusted + Web	Model 7
9 Single-T-I (Trusted)	Inception-v4	Trusted	Model 6
10 Single-T-IR (Trusted)	Inception-ResNet-v2	Trusted	Model 7
11 Single-T-I (Trusted + Web)	Inception-v4	Trusted + Web	Model 6
12 Single-T-IR (Trusted + Web)	Inception-ResNet-v2	Trusted + Web	Model 7

if it improves the model. However, if we were to compare both models in the same training iteration, the model initialised from the triplet network weights (Model 12b) did perform better than the model initialised on ImageNet weights (Model 5b). Among the different networks trained, the triplet network performed the worst, but this is because their evaluation methods differ from one another (Argmax vs dictionary similarity comparison). Moreover, they were trained significantly less than their single CNN counterpart. In addition, the triplet network relies on the image dictionary for inference. The classes with fewer image samples, especially those with a single sample, may not have provided enough features in the dictionary.

4. Submissions

We submitted nine runs to PlantCLEF 2022, and its details are tabulated in Table 7. Apart from Run 1, our submissions were constructed from the ensemble of various models. Run 7, which constitutes three single CNNs based on the Inception-v4 and Incpetion-ResNet-v2 architectures, and trained on the trusted and web data, performed the best among our submissions.

4.1. Results

The test set was evaluated based on the average Mean Reciprocal Rank (MRR) per species, also known as the Macro-Averaged Mean Reciprocal Rank (MA-MRR) score. Our best Run (7) achieved an MA-MRR score of 0.6078 and was the fourth-highest among the 45 submissions. On the other hand, the top submission scored an MA-MRR of 0.6269. The results of the overall submissions are summarised in Figure 3.

Comparing our Run 2 (trusted only) and Run 3 (trusted and web), the models trained on both trusted and web datasets performed better. Once again, showing that web or noisy data does improve the generalisation of deep learning as in PlantCLEF 2017 [12]. Furthermore, combining models of different architectures, i.e., Inception-v4 and Inception-ResNet-v2 did slightly boost

Table 5
Top-1 Accuracy of the Experimented Models on the Validation Dataset.

Model	Training steps	Class	Order	Family	Genus	Species
1 Single-I (Trusted)	352,497	0.9497	0.7489	0.7197	0.5819	0.3927
2 Single-IR (Trusted)	421,517	0.9601	0.7845	0.758	0.6388	0.462
3 Single-I (Trusted + Web)	530,964	0.9582	0.7792	0.7542	0.6431	0.4608
4 Single-IR (Trusted + Web)	522,583	0.9618	0.7921	0.7647	0.6456	0.4545
5a Single-IR (Trusted + Web)	142,478	0.9461	0.7055	0.6733	0.5295	0.3363
5b Single-IR (Trusted + Web)	60,000	0.9307	0.6342	0.5925	0.4192	0.2201
6 Triplet-I (Trusted)	91,294	-	-	-	-	0.1227
7 Triplet-IR (Trusted)	99,666	-	-	-	-	0.0998
8 Triplet-IR (Trusted + Web)	111,056	-	-	-	-	0.1318
9 Single-T-I (Trusted)	32,539	0.9252	0.6118	0.5733	0.3954	0.2057
10 Single-T-IR (Trusted)	207,971	0.9586	0.7567	0.7291	0.6024	0.4265
11 Single-T-I (Trusted + Web)	32,046	0.9248	0.6052	0.5635	0.3819	0.1898
12a Single-T-IR (Trusted + Web)	193,111	0.9567	0.748	0.7175	0.5839	0.3921
12b Single-T-IR (Trusted + Web)	60,000	0.9396	0.6712	0.6274	0.4566	0.2522

Table 6
Top-5 Accuracy of the Experimented Models on the Validation Dataset.

Model	Training steps	Class	Order	Family	Genus	Species
1 Single-I (Trusted)	352,497	0.9999	0.9302	0.8997	0.7788	0.562
2 Single-IR (Trusted)	421,517	1	0.9438	0.9143	0.8236	0.6491
3 Single-I (Trusted + Web)	530,964	1	0.9416	0.9143	0.8286	0.6517
4 Single-IR (Trusted + Web)	522,583	1	0.9456	0.9193	0.8329	0.6468
5a Single-IR (Trusted + Web)	142,478	1	0.9108	0.8695	0.7395	0.5236
5b Single-IR (Trusted + Web)	60,000	0.9999	0.877	0.8195	0.6414	0.3748
6 Triplet-I (Trusted)	91,294	-	-	-	-	0.232
7 Triplet-IR (Trusted)	99,666	-	-	-	-	0.1976
8 Triplet-IR (Trusted + Web)	111,056	-	-	-	-	0.2478
9 Single-T-I (Trusted)	32,539	0.9999	0.8705	0.8102	0.6185	0.358
10 Single-T-IR (Trusted)	207,971	1	0.9335	0.9001	0.7985	0.6159
11 Single-T-I (Trusted + Web)	32,046	0.9998	0.8657	0.8027	0.6056	0.3337
12a Single-T-IR (Trusted + Web)	193,111	1	0.9298	0.8941	0.7853	0.5861
12b Single-T-IR (Trusted + Web)	60,000	0.9999	0.8968	0.8448	0.6785	0.4203

the performance from 0.5461 (Run 1: Inception-v4 only) to 0.5536 (Run2: Inception-v4 and Inception-ResNet-v2). Since our triplet models and single CNN initialised on triplet weights did not saturate, it did not help the ensemble models. Consequently, Run 3 and 7 dropped in performance when added with the triplet or single CNN triplet initialised models, as observed in Run 5 and 8.

Table 7
Performance of our Submissions.

Run	Model	MRR
1	Single-IR (Trusted)	0.5461
2	Single-I (Trusted) + Single-IR (Trusted)	0.5536
3	Single-I (Trusted + Web) + Single-IR (Trusted + Web)	0.6058
4	Single-I (Trusted) + Single-IR (Trusted) + Single-I (Trusted + Web) + Single-IR (Trusted + Web)	0.6038
5	Single-I (Trusted + Web) + Single-IR (Trusted + Web) + Triplet-IR (Trusted + Web)	0.5989
6	Single-IR (Trusted) + Single-IR (Trusted + Web)	0.5887
7	**Single-IR (Trusted) + Single-I (Trusted + Web) + Single-IR (Trusted + Web)**	**0.6078**
8	Single-IR (Trusted) + Single-I (Trusted + Web) + Single-IR (Trusted + Web) + Single-T-IR (Trusted + Web) + Single-T-IR (Trusted)	0.6011
9	Single-IR (Trusted) + Single-IR (Trusted) + Single-I (Trusted + Web) + Single-IR (Trusted + Web)	0.603

5. Conclusion

We trained several Inception-v4 and Inception-ResNet-v2 single and triplet deep learning models for the plant identification task in PlantCLEF 2022. Due to its large number of species and training data, it was indeed resource-intensive to experiment. Due to resource and time constraints, our models were not rightfully saturated, and we did not experiment as intended. Therefore, we would like to look into the performance between our single CNN and our triplet models for future work. It is worth looking into their performance when they are both saturated and compared with the same evaluation methods and on different validation sets focusing on unbalanced classes. Nevertheless, we submitted nine runs and our best submission achieved a Macro Averaged Mean Reciprocal Rank score of 0.6078, placing 4th among the 45 submitted runs. In addition, we have shown that web or noisy data does improve generalisation in the identification. Furthermore, the ensemble of models from different network architectures, i.e., Inception-v4 and Inception-ResNet-v2, give higher accuracy than a single model.

Acknowledgments

The resources of this project is supported by NEUON AI SDN. BHD., Malaysia.

References

[1] H. Goëau, P. Bonnet, A. Joly, Overview of PlantCLEF 2022: Image-based plant identification at global scale, in: Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, 2022.

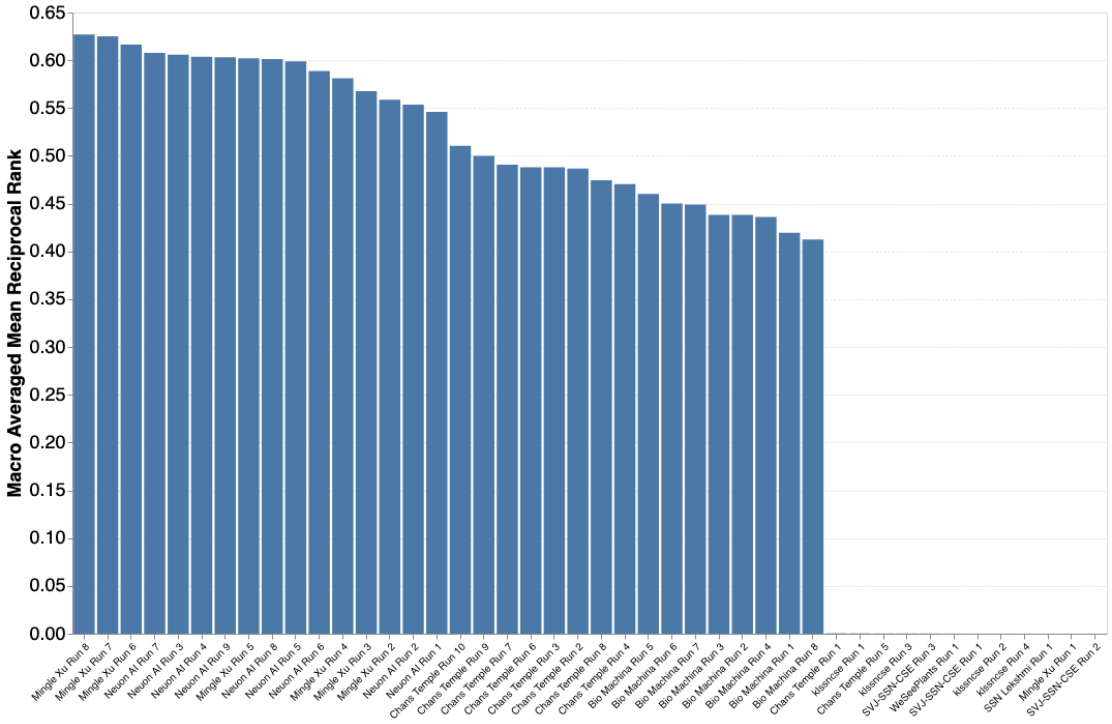

Figure 3: The Official Results of PlantCLEF 2022.

[2] A. Joly, H. Goëau, S. Kahl, L. Picek, T. Lorieul, E. Cole, B. Deneu, M. Servajean, A. Durso, H. Glotin, R. Planqué, W.-P. Vellinga, A. Navine, H. Klinck, T. Denton, I. Eggel, P. Bonnet, M. Šulc, M. Hruz, Overview of lifeclef 2022: an evaluation of machine-learning based species identification and species distribution prediction, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2022.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (2015) 211–252.

[5] H. Goëau, P. Bonnet, A. Joly, Overview of lifeclef plant identification task 2020, in: CLEF 2020-Conference and labs of the Evaluation Forum, 2020.

[6] H. Goëau, P. Bonnet, A. Joly, Overview of plantclef 2021: cross-domain plant identification, in: Working Notes of CLEF 2021-Conference and Labs of the Evaluation Forum, volume 2936, 2021, pp. 1422–1436.

[7] S. Chulif, Y. L. Chang, Herbarium-field triplet network for cross-domain plant identification. neuon submission to lifeclef 2020 plant., in: CLEF (Working Notes), 2020.

[8] S. Chulif, Y. L. Chang, Improved herbarium-field triplet network for cross-domain plant identification: Neuon submission to lifeclef 2021 plant., in: CLEF (Working Notes), 2021,

pp. 1526–1539.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: https://www.tensorflow.org/, software available from tensorflow.org.

[10] Sergio Guadarrama, Nathan Silberman, TensorFlow-Slim: A lightweight library for defining, training and evaluating complex models in tensorflow, https://github.com/google-research/tf-slim, 2016. URL: https://github.com/google-research/tf-slim, [Online; accessed 29-June-2019].

[11] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[12] H. Goëau, P. Bonnet, A. Joly, Plant identification based on noisy web data: the amazing performance of deep learning (lifeclef 2017), in: CLEF: Conference and Labs of the Evaluation Forum, 1866, 2017.