Non-local DenseNet for PlantCLEF 2019 Contest*

Dat Nguyen Thanh, Georges Quénot, and Lorraine Goeuriot

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble France
datnt.hust59@gmail.com, georges.quenot@imag.fr, lorraine.goeuriot@imag.fr

Abstract. Image-based plant identification is a promising tool constituting the automation of agriculture and environmental conservation as stated in. As an attempt to tackle the data deficient challenge in PlantCLEF 2019, the DenseNet architecture with competitive performance and relatively low number of parameters is augmented with a non-local block. A variety of data sampling schemes are also evaluated as a part of the work. The evaluation of the model and the methods is detailed in the content of the paper.

Keywords: DenseNet · Non-local block · Plant Identification.

1 Introduction

Various types of plants grow all around us, yet, little amongs us are plant experts. Indeed, knowing what plant available and where they are will be extremely helpful in pharmacy, from productional and academical perspective, environment protection. The rising of machine learning with artificial neural networks and convolutional neural networks which, are able to performs at near-human capability in image processing task, the popular use case of such technologies are for the automation of the task which human already excels: face recognition, image classification, etc. Still, it is would be highly beneficial if we can leverage these technologies in the task that human are yet to excel at in mass: Plant Identification.

The image-based plant identification can be formulated as a plant classification problem, where the input is an image containing the plant and the output is the id of the plant pre-defined by user. Formulating the problem of PlantCLEF contest as an image-classification task, the task itself in general has observed drastic improvement with the deep learning based methods, in the summarization of PlantCLEF 2017[2], it is shown that the best competitors have got over 90% accuracy using the aforementioned method. Notably, in the LifeCLEF 2018

* Supported by MRIM Team

contest[3], the are quite a number of software that achieved comparable accuracy to that of the top experts.

In this work, we present our proposed methods for the PlantCLEF 2019 [4] which is part of the LifeCLEF 2019 [1] which focus on 10,000 species from data deficient regions. The rest of this paper is structured as follows: Section 2 gives an overview of related works on automatic plant-identification in deep learning from previous contests, section 3 describe the proposed architecture for prediction, section 4 provides additional information on data augmentation and data sampling schemes and finally we conclude our works in section 5.

The source code and trained models are made available under the github link: `https://github.com/datvo06/PlantCLEF2019MRIM`.

2 Related Work

Ever since AlexNet[9] won the competition of ImageNet classification 2012, Convolutional Neural Networks(CNNs) has always been at the center of image classification. Following AlexNet, there have been three lines of research focusing on the CNNs: modifying the operations in the CNNs, dividing the networks into several sub-modules and make improvement on each of them, and finally, altering the information flow by adding connections.

Fine-tuning modules and adding auxiliary loss The inception model [12,13,11] follows the principle of repeating many carefully designed block of filter stacked horizontally (receive the same input and the output feature map are concatenated). Each time with new version of Inception Net, the authors often optimizing one of these blocks so that the number of computations, memory consumption, number of parameters can be optimized. The Inception-v1 is used for the baseline of PlantCLEF 2017, achieving the Top 1 accuracy of 0.513

Adding Residual connections One of the problem with original deep neural networks is that the more layers added, there more model prone to gradient vanishing. Various works have been proposed to amend this problem, (i.e LSTM [7] for sequenced input, highway network [7] which introduce a gated mechanism for ANNs), for convolutional neural network, residual additive connection proposed by [5] is one of them, the author later analyzed carefully the effect of the order of each Residual Block, resulting in [6], a modified version of the ResNet used in PlantCLEF 2017 [2], achieved the best score among non-ensemble runs with top 1 accuracy of 0.853.

Combining Inception and ResNet The inception design and the ResNet design has merged together, first in the Inception-ResNet design [11]. The network architecture still bases on the original principle of carefully designed block, the authors did this by adding the residual connection in a few variant of inception blocks. Inception-ResNet v2 achieved similar score to ResNet modified in the PlantCLEF 2017 [2] with MRR 0.847, Top 1 0.794, Top 5 0.913 and are used by the majority participants in PlantCLEF 2018.

Ensemble prediction The top performer of PlantCLEF 2017 [10] utilized ensemble prediction of multiple predictions with bagged averaging, the models used are ResNet, ResNeXt [10] and Inception-v1.

DenseNet As the residual connections has been proven to allow better gradient flow and performance boost to the convolutional neural networks, the DenseNets author[8] has tested the idea of densely connected layers. The model capable of achieving state-of-the-arts accuracy in classifying tasks with a relatively low number of parameters, making it a potentially good baseline for the data-deficient context. For this reason we choose the DenseNet as the baseline for the model.

Data Sampling Schemes To the best of our knowledge on data-sampling for training, there are little overlapping works with the strategies proposed.

3 Model Architecture

3.1 Non-local Networks

The non-local neural network [14] was proposed to solve the problem of limited information propagation from CNN and LSTM. The idea is to performs inter-pixels correlations from different position in the feature maps, leading to generate more power pixel-wise representation. The non-local operation, according to [14] is defined as:

$$\mathbf{y}_i = \frac{1}{C(\mathbf{x})} \sum_{\forall j} d(\mathbf{x}_i, \mathbf{x}_j)\, h(\mathbf{x}_j) \tag{1}$$

Where i is the index on the output feature maps (in space, time, or spacetime in the original case of video classification, annotation), j is the index on the input feature maps $\mathbf{x}$ and d computes the scalar representing the pairwise relationship between the entities in the items reside in these locations.
We shall see on the next section where the non-local block is added to the DenseNet baseline.

3.2 Adding Non-local operation to the DenseNet

The non-local operation is added between the output of the third dense block and the 1×1 channel-squashing convolution.

The non-local block was added after the third dense block for several reason:

- First, in the original introduction of the non-local block [14], multiple non-local position has been tested, of which, the best position is after the third Residual Convolution Block
- We have known based on the mechanism of self-attention, the non-local block performs pairwise dot product between two transformation of every pair of pixels on the grid. That why it is necessary to place a few convolution blocks before the non-local block so that the operation may potentially leverage informations from local neighbors.

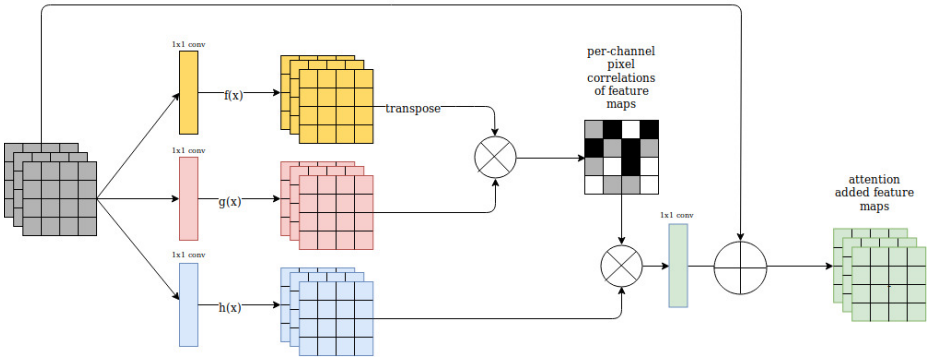


Fig. 1. Non-local block, $f(x), g(x), h(x)$ are three 1×1 convolutions, where $f(x), g(x)$ are channel-squashing functions.

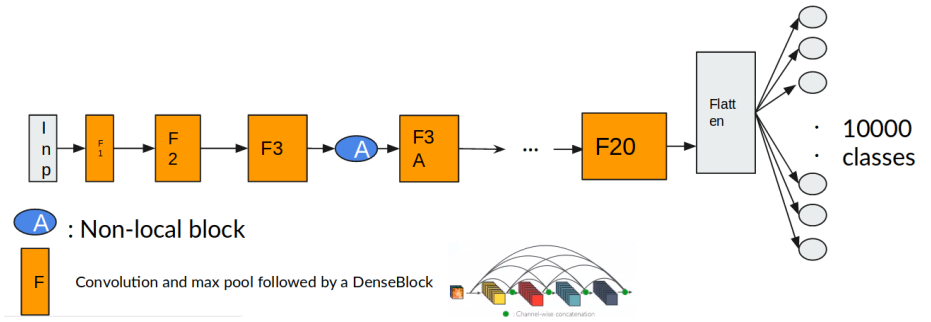


Fig. 2. Placing Non-local block within DenseNet.

3.3 Ensemble prediction

When applied into the final predictions, each instance of observation has multiple samples, so that there either has to be some middle layer to aggregate prediction in order to combine the prediction of multiple models on multiple instance. For this, a two-level pooling is leveraged:

The first level of pooling is used for ensembling the predictions of multiple different trained prediction instances and the second is used for aggregation of predictions from multiple observations.

4 Experiments and Results

4.1 Data Augmentation

Several data augmentation strategies have been applied:

– Randomly resizing image

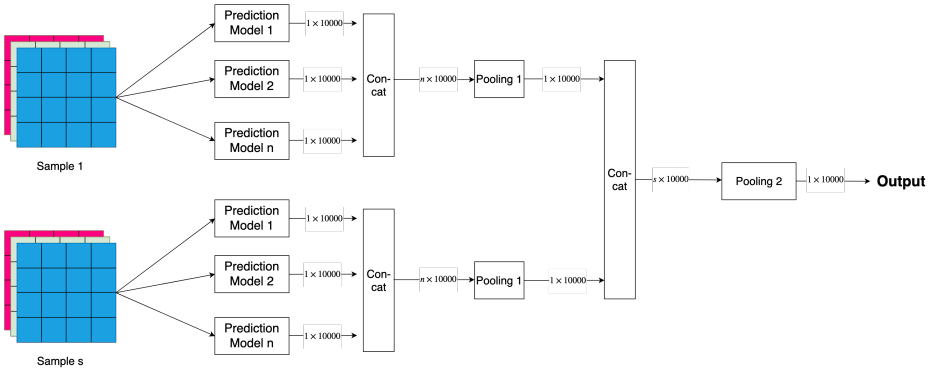


Fig. 3. Ensemble prediction using two-level pooling.

- Randomly crop
- Random Horizontal Flip
- Random alternating the brightness and contrast

Fig. 4. Illustration of data-augmentations.

4.2 Data Sampling

Notation:

- N: total number of samples
- n_i: number of samples for i^{th} class
- o_i: oversampling factor for i^{th} class
- w_i: sampling weight for i^{th} class
- m: the median number of samples
- μ: the mean number of samples per classes.

Minimum Threshold Resampling This strategy only focus on augmenting the classes having less number of samples than the average number of samples

per classes. Here, for each class with number of samples n_i, the oversampling factor o_i will be assigned the value of μ/n_i.
The oversampling might make some samples in the classes appears too many times compared to the others, making the model prone to overfit and also, so on each epoch, the classes samples are reshuffled and resampled.
Another problem is that the training times will be prolonged due to the increase in number of samples. For this, another strategy is also applied which is described below.

Smoothed Re-sampling This strategy partly oversampling small classes while also performs subsampling on classes with large number of samples. All of the aforementioned parameters are constant during training. The number of total samples which will be used throughout the training session is the sample: N. On each epoch however, each of the classes will be under-sampled or oversampled based on the weight w_i, total weight on one epoch will be normalized so that the number of total samples will always be equal to N: $\sum_i w_i o_i n_i$. We will now turn to how to choose the o_i and w_i factors. With the $m = 10$ for examples, all the classes will initially applied the oversampling factor o_i. The oversampling ensures a minimal number and diversity via data augmentation.

- $o_i = 1$ for $n_i > m$ (no oversampling beyond median).
- $o_i = (1 + m/n_i)/2$ for $n_i \leq m$ (oversampling for linear importance between $m/2$ and m).

Oversampling reduces the imbalance from about 1000:1 to about 100:1.
Weighting further ensures a better balancing using a power law.
$w_i = (o_i n_i)^{\gamma-1}$.

- With $\gamma = 1.0$, no weighting, original case (except the oversampling effect).
- With $\gamma = 0.5$, weighting further reduces the imbalance from about 100:1 to about 10:1.
- With $\gamma = 0.25$, weighting further reduces the imbalance from about 100:1 to about 3:1.
- With $\gamma = 0.15$, weighting further reduces the imbalance from about 100:1 to about 2:1.

In all cases, re-normalize (divide each w_i by the same value so that $\Sigma_i(w_i o_i n_i) = N$).

4.3 Experiment Results on the PlantCLEF 2017

All the candidate models have been trained on the PlantCLEF 2017 for preliminary testing before being used on the PlantCLEF 2019. The models are trained on the EOL set and tested on Web dataset with the data augmentation strategies mentioned in the subsection 4.1. The result is shown in the table 1.

It is can be easily seen that the Non-local addition added an increase of accuracy in both the DenseNet-121 and DenseNet-201 and the DenseNet slightly out performs the ResNet.

Table 1. Evaluation of trained models on PlantCLEF 2017 Web.

Model	Top 1 Accuracy
ResNet-18	0.5111
ResNet-152	0.7888
ShuffleNet	0.7222
DenseNet-121	0.8126
Non-local DenseNet-121	**0.8618**
DenseNet-201	0.8515
Non-local DenseNet-201	0.8744

4.4 Experiment results on PlantCLEF 2019

Initial result The model are further tested on the PlantCLEF 2019 dataset. The initial result is shown in Table 2. Thus, we can easily observe a drastic

Table 2. Model Performance on PlantCLEF 2019 Validation Set.

Model	Top 1 Accuracy
DenseNet-121	0.2510
DenseNet-201	0.3503
Non-local DenseNet-201	**0.4525**

performance drop. The further inspection of the dataset shows some challenging properties:

1. The classes are imbalanced
2. Repeated samples across the classes makes the learning harder.
3. Noisy Samples

Experiment Results on the Class-Filtered PlantCLEF 2019 We first test the effects of following strategies:

- Temporary removing all the classes with less than 5 samples
- Further remove noisy/incorrect formatted images.

The result is a 8500-classes dataset with still over 400,000 samples. The evaluation of the model is shown is Table 3. The result does not show much differences.

Experiment Results on the Repetition-Filtered PlantCLEF 2019 Further experiments are performed on the dataset with different thresholds for repetition, the following training/validation split strategy is applied: for each class, at least $\lceil n_{samples}/5 \rceil$ is taken as part of the validation set, if the class has only one samples, the training set for that class would be empty. Here, the minimum threshold sampling is applied. The evaluation result is shown in table 4.

Table 3. Model evaluation from small-class-filtered dataset.

Model	Top 1 Accuracy	Additional Condition
DenseNet-121	0.3020	None
DenseNet-201	0.4220	None
DenseNet-201	0.4890	Balanced Sampling
Non-local DenseNet-201	0.5215	Oversampling data-deficient classes

Table 4. Filtering out inter-class repeated samples makes training and validating set different.

Max repetitions	Number of empty classes	Top 1 Training	Top 1 Testing
1	1539	0.8425	0.1925
2	1040	0.6530	0.1512
3	778	0.6930	0.1451

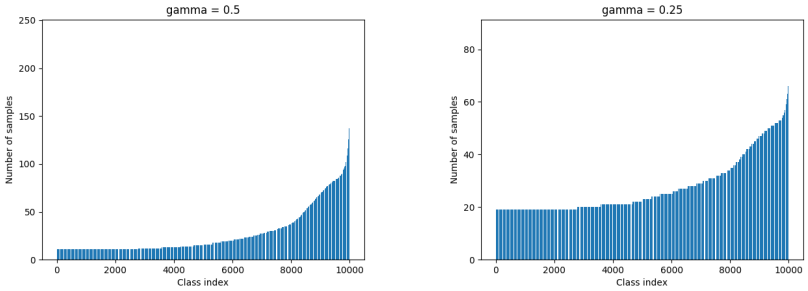
It can easily be seen that removing the all repetitions from duplication creates empty classes, which would heavily differentiates the training and validating set, making it hard to validate the model.

Experiment Results on conditional repetition filtered PlantCLEF 2019
On the final try of filtering the dataset involves filtering out all repeated samples unless it creates empty class. The statistics of the resulting dataset is stated in Table 5.

Table 5. Conditional Repetition Filtered PlantCLEF 2019.

Attributes of Dataset	Attribute Value
Number of classes	10,000
Number of samples	279,832
Mean number of samples	27.98
Minimum number of samples per class	1
Median number of samples	5
Max number of samples per class	1202
Number of unique samples	278,906
Number of samples duplicated	158

With all the repeated samples trimmed, the distribution is still pretty imbalanced, Figure 5 shows the distributions with Smoothed Resampling strategy. Since this is the final try, the whole dataset has to be used for training, for this, other external datasets has to be used for testing. More inspections on the PlantCLEF 2017 dataset reveals that there are 551 common categories betweens the PlantCLEF 2017 and PlantCLEF 2019 dataset. The samples are sorted by sizes and filtered to avoid having them in the training set. The statistic of the dataset is shown in table 6.



(a) $\gamma = 0.5$ (b) $\gamma = 0.25$

Fig. 5. Effect of different γ.

Table 6. Validation Dataset Statistics.

Dataset	PlantCLEF 2017 EOL Common	PlantCLEF 2017 Web Common
Number of classes	551	551
Number of samples	10,803	63,242

The final obtained results before submission testing on these dataset are described in the table 7:

Training Set	γ	No. instances	Pooling	2017 EOL	2017 Web	2017 EOL + Web
Conditional Filtered PlantCLEF 2019	0.25	4	Mean	0.9171	0.6635	0.6983
			Max	0.9169	0.6637	0.6984
	0.5	4	Mean	0.9455	**0.6970**	**0.7311**
			Max	0.9413	0.6941	0.7280
	1	2	Mean	0.9138	0.6476	0.6842
			Max	0.9011	0.6338	0.6705
	Mixed	10	Mean	**0.9478**	0.6957	0.7303
			Max	0.9371	0.6812	0.7163
All Data	No	1	No	0.7852	0.5497	0.5821

Table 7. Non-local DenseNet 201 Evaluation on PlantCLEF 2017 Common Categories.

We can see that with the same model, trained on the same number of epochs, the filtering strategies shows the differences: The ensemble of 4 model trained with $\gamma = 0.5$ gives of the best performance, the model which trained with all data from PlantCLEF 2019 is also evaluated and compared.

Final Test Results The final results are given by the top 1 accuracy on the test dataset and the hand-picked subset by experts. The detail of each run is

given in table 8. The best accuracy of top 1 on the expert-chosen samples set is achieved with the mean of 4 instances trained with $\gamma = 0.25$ with 2 means pooling, and best accuracy of top 1 on all samples is chosen with $\gamma = 0.5$ and two max pooling.

Run	γ	No. instances	Pooling 1	Pooling 2	Top 1 (expert)	Top 1 (all)	Top 3 (expert)
1	0.25	4	Mean	Mean	**0.043**	0.042	0.051
2	0.5	4	Mean	Max	0.017	0.036	0.043
3	0.25	4	Max	Mean	0.017	0.030	0.060
4	0.25	4	Max	Max	0.009	0.027	0.060
5	0.5	4	Mean	Mean	0.017	0.036	0.043
6	0.25	4	Mean	Max	0.017	0.028	0.051
7	0.5	4	Max	Mean	0.026	0.042	**0.085**
8	0.5	4	Max	Max	0.034	**0.046**	0.068
9	1	2	Mean	Max	0.017	0.031	0.043
10	Mixed	10	Mean	Max	0.026	0.034	0.068

Table 8. Final Run evaluation.

5 Conclusion

Plant Identification is an important step in medical, agricultural and environment resource planning. However, the problem is currently still a challenging to both human and computer vision-based technologies even with the development of deep learning. With data-deficient challenge, the problem is even harder to conquer. The work aims to provide a decent-performing model proven with extensive experiments along with a variety of data-handling strategies, yet it still cannot solve the whole problem. The remaining problems are avoiding of bias between classes belonging to the same genus, this perhaps can be performed by adding hierarchical classification where the system first identifies the genus and then the species. The data-deficient challenge still need to be tackled, either by leveraging unsupervised or semi-supervised learning methods. On the model designing perspective, the authors believe that the model can potentially be improved by adding inter-channel correlations in the non-local block.

References

1. Alexis Joly, Herv Goau, C.B.S.K.M.S.H.G.P.B.W.P.V.R.P.F.R.S.H.M.: Overview of lifeclef 2019: Identification of amazonian plants, south & north american birds, and niche prediction. In: Proceedings of CLEF 2019 (2019)
2. Goëau, H., Bonnet, P., Joly, A.: Plant identification based on noisy web data: The amazing performance of deep learning (LifeCLEF 2017). CEUR Workshop Proceedings **1866**(LifeCLEF) (2017)

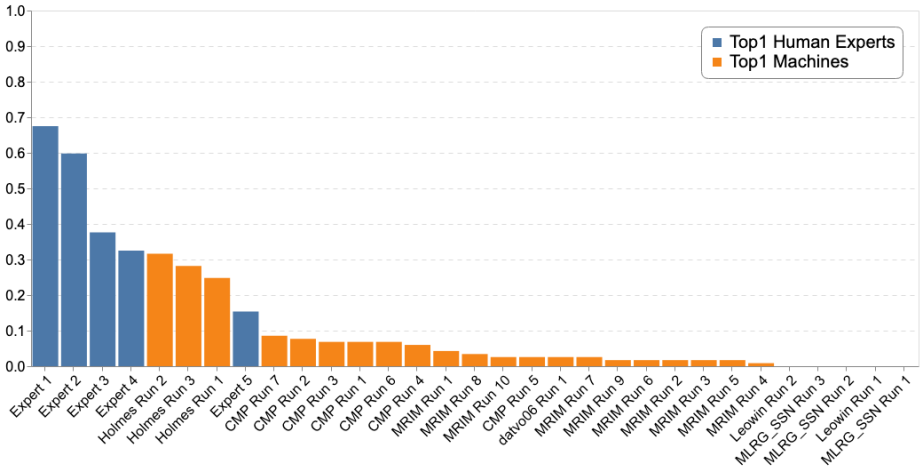


Fig. 6. Top 1 On Test Set, the best automatic solution performed at 0.316, while best experts have the accuracy of 0.675. Despite obtaining the accuracy of only 4.3%, the team is in top 3, the top performing model on the top 1 accuracy is the first run.

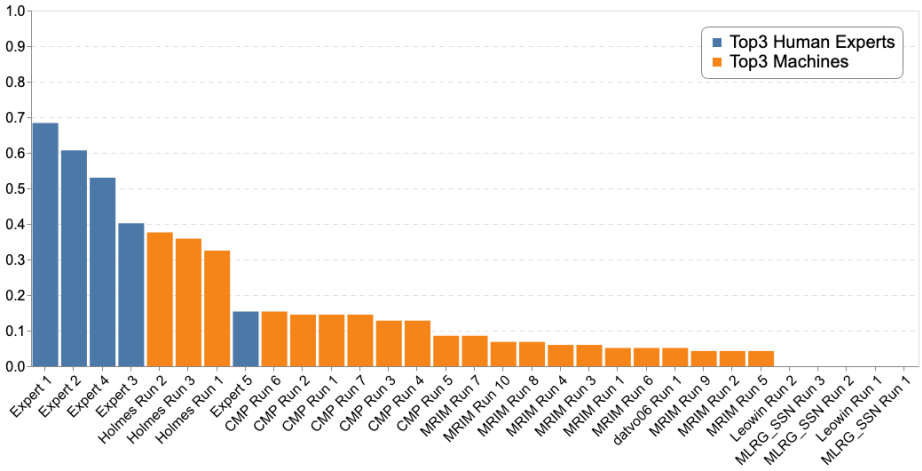


Fig. 7. Top 3 On Test Set.

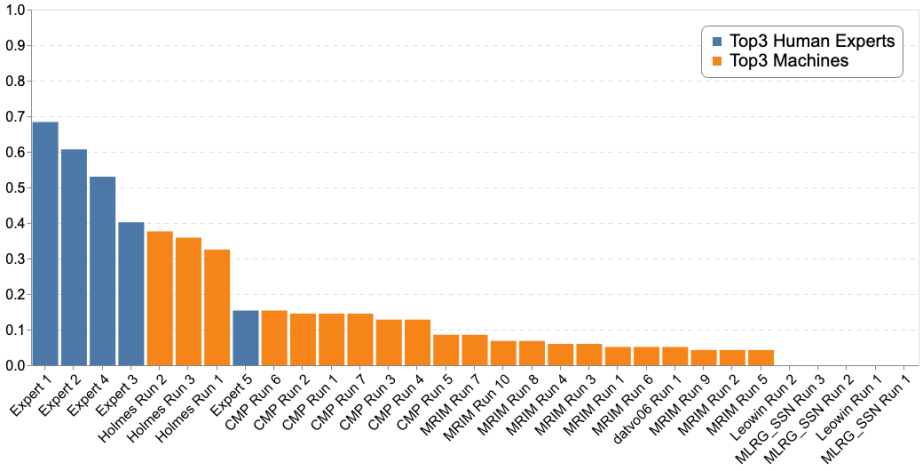


Fig. 8. Top 5 On Test Set.

3. Goëau, H., Bonnet, P., Joly, A.: Overview of ExpertLifeCLEF 2018: How far automated identification systems are from the best experts? CEUR Workshop Proceedings **2125** (2018)
4. Goëau, H., Bonnet, P., Joly, A.: Overview of lifeclef plant identification task 2019: diving into data deficient tropical countries. In: CLEF working notes 2019 (2019)
5. 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). pp. 770–778. IEEE (jun 2016). https://doi.org/10.1109/CVPR.2016.90, `http://ieeexplore.ieee.org/document/7780459/`
6. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **9908 LNCS**, 630–645 (2016). https://doi.org/10.1007/978-3-319-46493-0_38
7. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, `http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735`
8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 (2017). https://doi.org/10.1109/CVPR.2017.243
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Network. Proceedings of the 25th International Conference on Neural Information Processing Systems **1**, 1097—-1105 (2012). https://doi.org/10.1061/(ASCE)GT.1943-5606.0001284, `http://dl.acm.org/citation.cfm?id=2999134.2999257`
10. Lasseck, M.: Image-based plant species identification with deep Convolutional Neural Networks. CEUR Workshop Proceedings **1866** (2017)
11. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning (2016).

https://doi.org/10.1016/j.patrec.2014.01.008, http://arxiv.org/abs/1602.07261
12. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going Deeper with Convolutions pp. 1–12 (sep 2014), http://arxiv.org/abs/1409.4842
13. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision (dec 2015), http://arxiv.org/abs/1512.00567
14. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local Neural Networks. Tech. rep. (2018)