THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Leveraging multiple datasets for deep leaf counting

Citation for published version:
Dobrescu, A, Giuffrida, MV & Tsaftaris, S 2018, Leveraging multiple datasets for deep leaf counting. in *International Conference on Computer Vision: Computer Vision Problems in Plant Phenotyping (workshop).* Institute of Electrical and Electronics Engineers (IEEE), International Conference on Computer Vision, Venice, Italy, 22/10/17. https://doi.org/10.1109/ICCVW.2017.243

Digital Object Identifier (DOI):
10.1109/ICCVW.2017.243

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
International Conference on Computer Vision

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 27. Jul. 2024

Leveraging multiple datasets for deep leaf counting

Andrei Dobrescu
University Of Edinburgh
A.Dobrescu@ed.ac.uk

Mario Valerio Giuffrida
IMT Lucca
valerio.giuffrida@imtlucca.it

Sotirios A Tsaftaris
University Of Edinburgh
S.Tsaftaris@ed.ac.uk

Abstract

The number of leaves a plant has is one of the key traits (phenotypes) describing its development and growth. Here, we propose an automated, deep learning based approach for counting leaves in model rosette plants. While state-of-the-art results on leaf counting with deep learning methods have recently been reported, they obtain the count as a result of leaf segmentation and thus require per-leaf (instance) segmentation to train the models (a rather strong annotation). Instead, our method treats leaf counting as a direct regression problem and thus only requires as annotation the total leaf count per plant. We argue that combining different datasets when training a deep neural network is beneficial and improves the results of the proposed approach. We evaluate our method on the CVPPP 2017 Leaf Counting Challenge dataset, which contains images of Arabidopsis and tobacco plants. Experimental results show that the proposed method significantly outperforms the winner of the previous CVPPP challenge, improving the results by a minimum of 50% on each of the test datasets, and can achieve this performance without knowing the experimental origin of the data (i.e. "in the wild" setting of the challenge). We also compare the counting accuracy of our model with that of per leaf segmentation algorithms, achieving a 20% decrease in mean absolute difference in count ($|DiC|$).

1. Introduction

Plant phenotyping is a growing field that biologists have identified as a key sector for increasing plant productivity and resistance, necessary to keep up with the expanding global demand for food. Computer vision and machine learning are important tools to help loosen the bottleneck in phenotyping formed by the proliferation of data generating systems without all the necessary image analysis tools [22].

The number of leaves of a plant is considered one of the key phenotypic metrics related to its development and growth stages [28, 30], flowering time [18] and yield potential. Automated leaf counting based on imaging is a difficult task. Leaves vary in shape and scale, they can be difficult to distinguish and are often occluded. Moreover, a plant is a dynamic object with leaves shifting, rotating and growing between frames which can be challenging to computer vision counting approaches [22].

From a machine learning perspective, counting the number of leaves can be addressed in two different ways: (i) obtaining a per-leaf segmentation, which automatically leads to the number of leaves in a rosette [24, 26, 27]; or (ii) learning a direct image-to-count regressor model [11, 23]. Deep learning approaches in this field show impressive results in obtaining leaf count as a result of per-leaf segmentation, but they require individual leaf annotations as training data, which are difficult and laborious to produce. In fact, regression approaches leverage this issue by using the total leaf count in plants as its only supervision information [24, 26]. There are few annotated datasets for rosette plants [3, 5, 21] which is a limitation when trying to implement deep learning approaches for plant phenotyping problems [29] or for field of phenotyping in general [20].

In this paper, we propose a deep learning model for leaf counting in rosette plants on top-down view images. The backbone of the model is a modified *Resnet50* deep residual network [14] pre-trained on the ImageNet dataset (c.f. Figure 1). The network is fine-tuned on one or more datasets and provides as output a leaf count. To boost deep learning performance to learn despite being provided with small datasets, we found that pooling data from different sources and even different species (and cultivars) for the purposes of training improves leaf prediction accuracy. Our method treats leaf counting as a direct regression problem, therefore it only requires the total leaf count of each image as annotation. We evaluate our approach on datasets provided in the *Leaf Counting Challenge* (LCC) held as part of the *Computer Vision Problems in Plant Phenotyping* (CVPPP 2017) workshop. The datasets consist of top-down images of single plants of Arabidopsis (A1, A2, A4) and tobacco (A3) plants collected from a variety of imaging setups and labs. In this challenge, there was also a "wild" test dataset (A5) which combines test images from all the other datasets in order to assess the generalization capabilities of machine

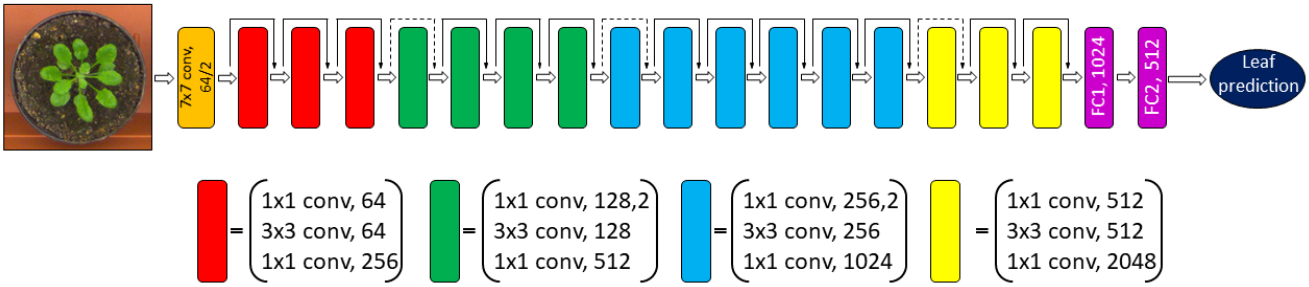

Figure 1. Architecture of our modified *Resnet50* model [14]. The network takes as input a RGB image of a rosette plant and outputs a leaf count prediction. The classification layer was removed and replaced with two fully connected layers FC1 and FC2. The network is comprised of 16 residual blocks which consist of three stacked layers each with skip connections between the input and the output of each block. The solid lines represent connections which maintain dimensions while dotted lines increase dimensions.

learning algorithms without knowing the experimental origin of the image data. The final model is robust to nuisance variability (i.e. different backgrounds, soil) and variations in plant appearance (i.e. mutants with altered plant shape and scale). We employed an ensemble method of four models to obtain the results of the LCC challenge. Experimental results show that our approach outperforms the winner of the previous CVPPP challenge [11] as well as a state-of-the-art counting via segmentation approach.

The remainder of this paper is organized as follows. In Section 2 we review the current literature. In Section 3 we detail our deep learning network. Then, in Section 4 we report the experimental results. Particularly, the results of the CVPPP challenge on the testing set are reported in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Counting via detection. One class of approaches involves counting by detection [31] which frames the problem as a detection task. Some solutions rely on local features such as histogram orientated gradients [4] or shape [6]. Object detectors using region based convolutional neural networks [9] have attracted attention by providing state-of-the-art detection results while reducing training and testing times. They incorporate region proposal [10, 25] and spatial pyramid pooling networks [13] to provide region of interest suggestions and then fine-tuning the resulting bounding boxes to fit on the desired objects.

Counting via object segmentation. Object detection is considered an easier task than segmentation. In fact, once an object is detected in the scene, obtaining a per-pixel segmentation mask is not trivial. Especially for the case of multi-instance segmentation [26], where the same objects appear multiple times in an image (e.g., leaves of a plant). Pape and Klukas [23] used split points to determine lines between overlapping leaves to assign them different labels. Deep learning solutions using recurrent neural networks achieve state-of-the-art leaf segmentation and counting results. In [26], the authors developed an end-to-end model of recurrent instance segmentation by combining convolutional LSTM [7] and spatial inhibition modules as a way to keep track of spatial information within each image allowing to segment one leaf at a time. The method also deploys a loss function which learns to segment all separate instances sequentially and allows the model to learn and decide the order of segmentation. In [24], another neural network that uses visual attention to compute instance segmentation jointly with counting was proposed. This method has sequential attention by creating a temporal chain via a LSTM cell which outputs one instance at a time. Non-maximal suppression, used to solve heavily occluded scenes, is dynamically leveraged using previously segmented instances to aid in the discovery of future instances.

Counting via density estimation. Another method to count objects in an image is by estimating their distribution, using local features. In [19], the authors have developed a loss function which aims to minimize Maximum Excess over SubArrays (MESA) distance. Other methods include density estimation by per-pixel ridge and random forest regression. Similar approaches can be found in [1, 2, 8], where regressors are used to infer local densities. However, these approaches are difficult to use for leaf counting, as they are challenged by the huge scale variability of leaves, as well as heavy occlusions and overlaps.

Direct count. Leaf counting results using machine learning solutions have been reported in past CVPPP challenges as well as in other independent reports which have identified plant datasets as compelling ways to test models. The winner of the previous CVPPP challenge [11] adopted a direct regression model through support vector regression. The method involved converting the image into a log-polar coordinate system before learning a dictionary of image features in an unsupervised fashion. The features were learned only

in regions of interest determined by texture heuristics. The use of the log-polar domain provided the method with rotation and scale invariance, however the scale of the leaves is an important feature to learn, as is can vary considerably within a plant and is directly correlated with the growth stage of the plant. In [23], the authors used a set of geometrical features to fit several classification and regression models. Using different tools available in WEKA [12], they found that the Random Subspace method [16] could obtain lowest |DiC| only using geometrical features.

3. Methodology

We implemented a deep learning approach for counting leaves in rosette plants. We used a modified *ResNet50* [14] residual neural network to learn a leaf counter taking as input a top-down view RGB image of a rosette plant. For this paper, we have adhered to the guidelines and data provided for the CVPPP leaf counting challenge.

3.1. The Network

The architecture of our model is displayed in Figure 1. We used a *Resnet50* because of its ability to generalize, which was crucial for this challenge for its "in the wild" setting, as well for its fast training and convergence speed. The ResNet architecture is easier to optimize than other deep networks and addresses the degradation problem present in very deep networks which states that as deep networks converge and accuracy gets saturated, it starts to degrade [14]. The problem is addressed by the residual convolutional blocks which make up the network which is described as follows. If $H(x)$ is an underlying mapping of several stacked layers and we assume that the layers can approximate a complex function then we can assume that they can also approximate the residual function $F(x) = H(x) - x$. This changes the original function from $H(x)$ to $F(x) + x$. Furthermore, identity mappings, implemented as skip connections, ease optimization because they help propagate the error gradient signal faster across all layers. The positive impact on optimization and learning grows with increased layer depth. [15]. The residual functions are learned with reference to the layer inputs facilitated by the skip connections between the residual blocks as seen in Figure 1.

We modified the reference *ResNet50* by removing the last layer intended for classification, flattening the network and adding two fully connected layers FC1 and FC2 of 1024 and 512 nodes respectively followed by ReLU activations. We apply an L2 activation regularization on the FC2 layer to penalize the layer activity during training and prevent overfitting. FC2 goes into a fully connected layer containing a single node which acts as the leaf count prediction.

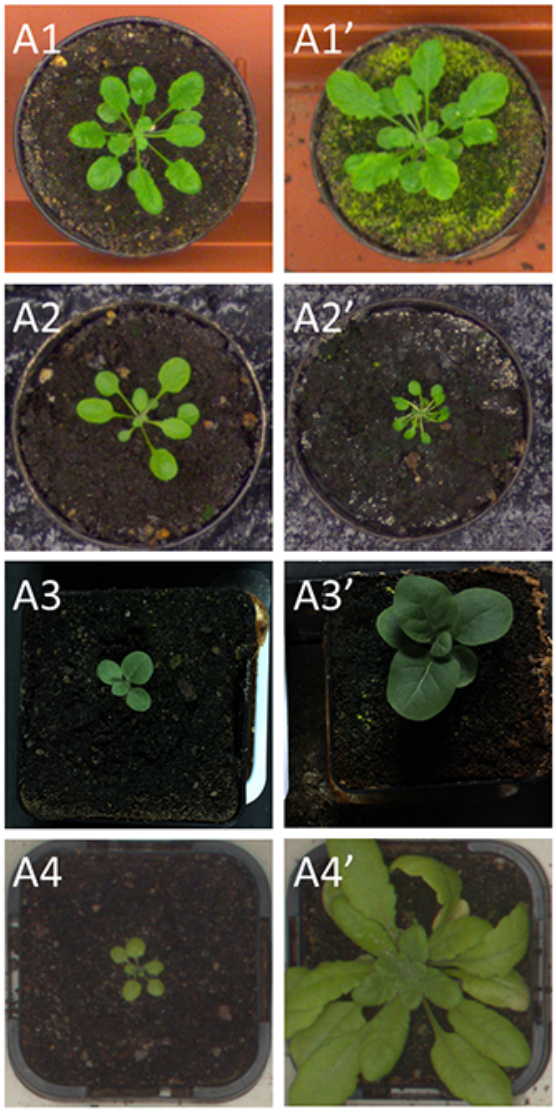

Figure 2. Examples of rosette plant images present in the four training datasets A1, A2, A3, and A4. The datasets show the big variety between images for applications in plant phenotyping. The left column represents images which have more well defined and easier to identify leaves, making leaf count relatively easier to determine. The right column shows examples of more challenging images for computer vision applications due to difficult to distinguish backgrounds (A1'), mutants which alter plant appearance (A2') and heavily occluded leaves (A3', A4').

3.2. The datasets

The challenge consisted of four RGB image training datasets noted as A1, A2, A3 and A4 [3, 21]. A1 and A4 contain images of wild-type Arabidopsis plants (Col-0) while A2 contains four different mutant lines of Arabidopsis which alter the size and shape of leaves. A3 is made up of young tobacco plant images. The training sets

include 128, 31, 27, 624 images and the testing sets contain 33, 9, 65, 168 images for A1, A2, A3, A4 respectively. The datasets are taken from different labs, with different experimental setups, and thus background appearance and genotype composition varies (Figure 2). Furthermore, the images in each dataset have different dimensions ranging from 500x530 pixels in A1 to 2448x2048 pixels in A3. For testing, in addition to the four aforementioned datasets, the organisers provided a "wild" dataset A5 which combines images from all testing datasets, to test methods that generalize across data and which are not fine-tuned (and specific) for each dataset. For training, we also formed a dataset similar to the "wild" dataset, named Ac, in which we combined all the images in the four training datasets.

3.3. Training procedure

For pre-processing, each image was re-sized to be 320x320x3 pixels and a histogram stretch was applied on all images to improve contrast as some images were darker than others. The resolution was chosen to optimize training times while retaining important features such as distinguishable small leaves. We used a random split from the training set to 50% of the images used for training 25% for validation and 25% for (internal) testing. The split ratio was chosen so that there would be a similar percentage of test images per run as the test set of the challenge. Furthermore, the training, validation and test sets include plants of all ages by taking an even distribution from each dataset. To evaluate our architecture, we first trained the network on each of the CVPPP datasets individually. We then added more data by combining datasets together (i.e. A1 + A2) and finally a combination of all four datasets named Ac. The test results from the combinations we evaluated can be seen in Table 1. Cross validation was performed four times on differently sampled training images when training on (Ac). We used mean squared error (MSE) as the loss function and Adam optimizer [17] with a learning rate of 0.001. We trained with an early stopping criterion, based on the validation loss to avoid overfitting.

Data augmentation was performed when training all models. We used a generator which assigns training images a random affine transformation from a pool of random rotation from 0-170 degrees, zoom between $0 - 10\%$ of the total image size and flipping vertically or horizontally. The training steps for each epoch was defined as the double the number of training images and the batch size per step was 6. In total, the augmented dataset was 12 times the size of the original set per training epoch. Once the models are trained, obtaining test predictions just requires inputting the desired test images. The network output is not discrete, so we round the predictions to the nearest integer to get a leaf count.

For the challenge test set, we employed an ensemble method comprised of four models trained on four different equal portions of the Ac dataset. We fused the predictions of the four models by averaging them to obtain the results shown in Table 3.

(A) Test results on training set A1

Train Sets	DiC	|DiC|	MSE	R^2	[%]
A1	-0.81(0.85)	0.94(0.70)	1.38	0.23	25
A1+A2	-0.06(1.03)	0.75(0.71)	1.06	0.76	41
A1+A4	-0.75(0.90)	0.88(0.78)	1.38	0.69	34
Ac	0.28(0.80)	0.53(0.66)	0.72	0.60	56

(B) Test results on training set A2

Train Sets	DiC	|DiC|	MSE	R^2	[%]
A2	-2.38(2.69)	2.38(2.69)	12.88	0.29	38
A1+A2	-0.56(2.06)	1.69(1.51)	6.31	0.65	25
A2+A4	-0.75(2.15)	1.75(1.45)	6.38	0.65	31
Ac	-0.38(1.11)	0.88(0.78)	1.38	0.92	38

(C) Test results on training set A3

Train Sets	DiC	|DiC|	MSE	R^2	[%]
A3	-0.57(1.50)	1.43(0.73)	2.57	0.46	14
Ac	0.71(1.03)	0.71(1.03)	1.57	0.47	57

(D) Test results on training set A4

Train Sets	DiC	|DiC|	MSE	R^2	[%]
A4	0.1(1.14)	0.91(0.85)	1.54	0.96	35
A1+A4	-0.01(1.06)	0.77(0.73)	1.12	0.97	39
A2+A4	0.05(1.04)	0.73(0.75)	1.10	0.97	43
Ac	0.12(0.99)	0.69(0.73)	1.01	0.97	46

Table 1. Evaluation results of models tested on just the training datasets. The first column of each table represents training regimen of our network comprising of single and combined datasets (Ac denotes a combination of all datasets). The values were obtained through cross-validation using a split of 50%, 25%, 25% images for training, validation and testing respectively. Test sets all refer to our internal split of the original training set as described in text.

3.4. Implementation details

We implement our models in Python 3.5. For training, we used an Nvidia Titan X 12Gb GPU using Tensorflow as backend. The models took between 1.5 and 5 hours to train depending on how many datasets were pooled together, over an average of 50 epochs.

3.5. Evaluation metrics

We used the same evaluation metrics as those provided by the organizers of the workshop to assess our networks performance: Difference in Count (DiC), absolute Difference in Count |DiC|, mean squared error (MSE) and percent agreement given by the percentage of exact predictions over total predictions. For our internal testing, we also include the R2 coefficient of determination.

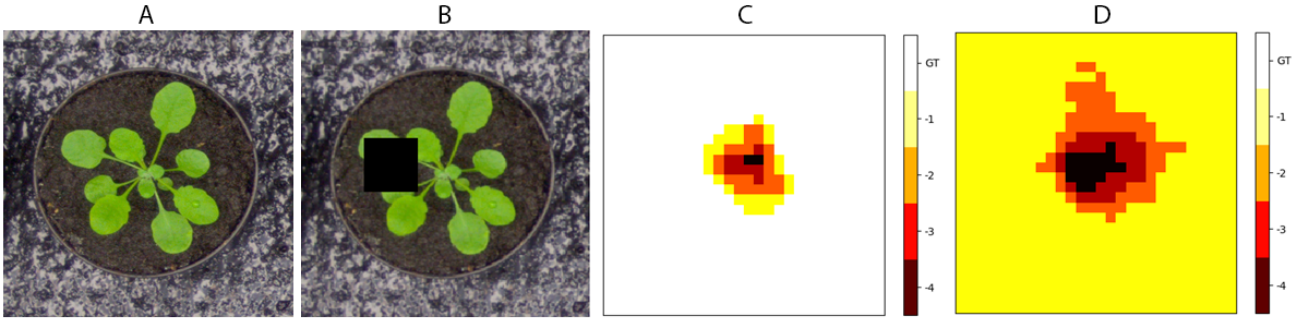

Figure 3. Test showing network training ability by obscuring part of the image with a sliding window. (A) is the original image and (B) shows the black sliding window (60x60 pixels) traversing the original image. (C) represents a heatmap of the accuracy of leaf count prediction as the sliding window is traversing the image using a model trained on the Ac dataset, showing that the errors are confined only to the area where the plant is located. (D) is the prediction heatmap on a model trained on just the A2 dataset (the origin of the image).

4. Results on the training set and discussion

We first tested our architecture on the training datasets in order to assess its performance and the results can be seen in Table 1. The tests were performed following the same training regimen outlined in the Section 3. We found that fine tuning the *Resnet50* network, pretrained on the ImageNet dataset, gave better and more consistent results than providing stronger annotations and using random initialization. So the learned ImageNet features were more valuable for this task than having the segmentation mask as an input.

Our solution was able to overcome several challenges. Firstly, the network can work with images of different sizes and scales present in each dataset. That said it is important that the original quality of the images is good enough to distinguish between leaves. Secondly, our model was able to learn to count leaves of different shapes, sizes and orientation provided only with minimal annotations. As seen in Figure 2, the plants were of a diverse nature in terms of age, genotype and species giving a wide degree of complexity in counting. This ties in with one of the limitations of the direct regression approaches, namely that the network has to infer more information from each image to compensate for the lack of stronger annotations. This can be attenuated by providing more data when training to give a better chance of learning relevant features. Labeling data is increasingly time and resource intensive when going from weak to stronger annotations (e.g total leaf count vs. per leaf segmentation mask) which is one of the reasons for the relative lack of publicly available plant phenotyoing datasets. By employing models which require only weak annotations for training as opposed to models which require strong annotations [24], it becomes possible to have access to more labelled data given limited resources. Lastly, the provided datasets contained a limited amount of training images compared to the quantities of data traditionally employed for deep learning models. Furthermore, they were not uniformly represented with set A2 including just 31 images, while set A4 had 623. Even with data augmentation, there was a barrier to how much the network would learn when trained on each dataset separately, as evident in Table 1. We sought to improve those initial results by combining the datasets together when training, so to provide more varied and meaningful examples than the ones supplied solely by data augmentation.

	DiC	|DiC|	MSE	R^2	[%]
A1*	-5.46(0.83)	5.46(0.82)	30.59	0.07	2
A1	-1.17(0.57)	1.17(0.57)	1.70	0.38	8
A1+A4*	-0.58(0.59)	0.59(0.59)	0.69	0.98	46

Table 2. Internal test metrics from models with and without (*) augmentation. The sets indicated with a * in the name (e.g. A1*) were trained without using data augmentation.

Results show that combining datasets from different sources and even species is beneficial, since it improves test accuracy for all datasets and more generally for all evaluation metrics (Table 1). The models were trained using data augmentation procedures mentioned above. The worst performance is seen in networks which are trained only on a single dataset. The combination of any two datasets yields similar results, no matter which combination is used even though A4 is a larger dataset than A2 and A1. The best results are given by the models trained on the grouping of all datasets (Ac). The degree of the improvement varies, with the most accentuated being A2 and the least impacted being A4. That was not unexpected, as A2 is the smallest dataset so additional data would cause a large impact in learning while A4 is by far the largest dataset. However, even in A4, the MSE and mean absolute difference in count (|DiC|) decreases by 30% when training on combined datasets com-

	DiC			|DiC|				Agreement [%]		MSE	
	Ours	Ref[11]	Ref[26]	Ours	Ref[11]	Ref[26]	Ref[24]	Ours	Ref[11]	Ours	Ref[11]
A1	-0.39(1.17)	-0.79(1.54)	0.2(1.4)	0.88(0.86)	1.27(1.15)	1.1(0.9)	0.8(1.0)	33.3	27.3	1.48	2.91
A2	-0.78(1.64)	-2.44(2.88)	-	1.44(1.01)	2.44(2.88)	-	-	11.1	44.4	3.00	13.33
A3	0.13(1.55)	-0.04(1.93)	-	1.09(1.10)	1.36(1.37)	-	-	30.4	19.6	2.38	3.68
A4	0.29(1.10)	-	-	0.84(0.76)	-	-	-	34.5	-	1.28	-
A5	0.25(1.21)	-	-	0.90(0.85)	-	-	-	33.2	-	1.53	-
All	0.19(1.24)	-	-	0.91(0.86)	-	-	-	32.9	-	1.56	-

Table 3. Results for the test set for the provided datasets based on the fused ensemble of models trained on dataset Ac. For comparison we include the results of the winner of the previous CVPPP leaf counting challenge [11] as well as two count derived from per leaf segmentation approaches [26, 24].

pared to just on A4 (Table 1D).

There was also a significant improvement between models trained on just A3 compared to Ac (Table 1C), especially in terms of |DiC| reducing it by 50%, MSE by 40% and percent agreement improved 4-fold. As the models trained on just A3 and Ac were presented with the same number of tobacco plant images, we see that the Ac model shows improved learning even when the only extra data comes from other plant species.

One natural question that arises is whether the network is learning to actually look at leaves to count or is influenced by the material in the background (i.e. it relies on background cues). We tested the network's ability to learn by imposing a black sliding window on images used in training [32]. We predicted the leaf count using our models on the images as the sliding window was traversing the image to see if by obstructing the portion of the image the network was meant to learn it would give rise to errors in count. By sliding across the image and reporting a result (of prediction error) per position of the sliding window, a heat map of errors is generated. An example of this process is shown in Figure 3. We used the sliding scale test on models trained on the aggregated dataset Ac (Figure 3C) and on models trained on just the dataset which contained the original image (Figure 3D). The errors in the model trained on dataset Ac were very specific to regions containing the plant suggesting that the model learned well the area of interest. As the sliding window moved closer to the center of the plant the errors increase as the window obstructs smaller leaves. The model trained on only one dataset did not perform as well, having a 1 difference error in general (yellow homogeneous background) and the regions which were affected by the sliding box were not only specific to the plants location.

We believe that the improved results are a consequence of increased data variability which allows the model to learn more precise features. To rule out that improved results could also be reached just by additional image augmentation we tested a combined trained model of A1+A4 with no augmentation versus a model of just A1 with data augmentation that would bring the total number of training images to be equal to that of A1+A4. The dataset A1 was chosen for this experiment because it is the most balanced dataset in terms of size. The results show a decreased test MSE of by more than 100% and the R^2 coefficient went from 0.38 to 0.98 (Table 2) in the models trained on the three datasets without augmentation compared to the single A1 with augmentation.

5. Results on the testing set as provided by the challenge organizers and discussion

The results of the LCC were compiled by the organizers using the metrics provided (Table 3). We did not use the provided segmentation masks or the leaf center dots in our models. We used an ensemble method comprised of four models of the same architecture but trained on different equal portions of the Ac combined dataset. We ran each of the models on the test set and fused the leaf count predictions by averaging the outputs to reach the results in Table 3. The A3 image dataset is an important indicator of how well a model learned to generalize because it contains 27 images in the training set and 65 images in the testing set while having the only tobacco images in the challenge.

We outperformed the winner of the previous CVPPP LCC by at least 50% for each of the datasets provided in MSE and |DiC| without needing to know the experimental origin of the data (i.e. "in the wild" setting of the challenge). We achieved an average of 33% agreement across the datasets with the exception of A2 where we had lower percent agreement than than the reference although our approach resulted in a significantly lower MSE. Training our models on the combined set Ac helped us achieve similar or better results for the A5 "wild" dataset compared to that of individual datasets.

The previous challenge only provided datasets A1, A2 and A3 to the participants, thus we can only compare results on those three datasets. Overall, we had an improvement on all parameters for all three datasets, with the largest improvements in A2, which represented the smallest Arabidopsis dataset and contained difficult mutants with altered shape and size.

We also compared our results with the state-of-the-art per leaf segmentation approaches from [26] and [24]. We outperformed [26] by achieving a 20% reduction in mean |DiC| compared to them. We also obtained similar results to [24], with the advantage of requiring less strong annotations compared to their method.

6. Conclusions

Here we presented a deep learning approach for leaf counting in rosette plants from top-down view RGB images. We showed that it is possible to use deep learning as a way of directly performing automated leaf counting for plant phenotyping and improve upon the current state of the art. Firstly, we implemented a modified ResNet50 deep residual neural network to act as a leaf prediction model where we treated leaf counting as a direct regression problem using only the total leaf count per plant as required annotation. Secondly, we found that pooling data from different sources for the purposes of training improves leaf prediction accuracy. Importantly, we also discovered that training on aggregated datasets also provides the model invariance to plant species, which is an important factor when testing the model with the "wild" set.

We evaluated our network on the standardized datasets provided in the context of the Leaf counting challenge of the CVPPP 2017 workshop and then compared to previously published networks. We found that our method outperforms the previous winner of the challenge by at least 50% for all the provided datasets. Furthermore, we compared our results with count from per leaf segmentation approaches and we achieved improved and comparable results to the two state-of-the art solutions currently available. In the first case, we outperformed the state-of-the-art network by decreasing the mean |DiC|, and in the second case we required far less strong annotation to achieve comparable results.

For the purposes of this challenge we only had access to images provided, so we could not investigate or optimize other parameters such as ideal camera placement for this task, but it would be an interesting avenue to pursue.

7. Acknowledgements

This work was supported by an EPSRC DTP PhD fellowship, number EP/N509644/. We acknowledge and thank NVIDIA for providing hardware essential for our work. Finally we would like to thank all the organisers of the CVPPP workshop for making it happen.

References

[1] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Interactive Object Counting. pages 504–518, 2014.

[2] C. Arteta, V. Lempitsky, and A. Zisserman. Counting in the Wild. 1:483–498, 2016.

[3] J. Bell and H. Dee. Aberystwyth Leaf Evaluation Dataset, 2016.

[4] A. Chayeb, N. Ouadah, Z. Tobal, M. Lakrouf, and O. Azouaoui. Hog based multi-object detection for urban navigation. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2962–2967, 2014.

[5] J. A. Cruz, X. Yin, X. Liu, S. M. Imran, D. D. Morris, D. M. Kramer, and J. Chen. Multi-modality imagery database for plant phenotyping. *Machine Vision and Applications*, 27(5):735–749, 2016.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] L. Fiaschi, R. Nair, U. Koethe, and F. A. Hamprecht. Learning to Count with Regression Forest and Structured Labels. In *21st International Conference on Pattern Recognition (ICPR 2012)*, pages 2685–2688, 2012.

[9] R. Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[11] M. V. Giuffrida, M. Minervini, and S. Tsaftaris. Learning to Count Leaves in Rosette Plants. In *CVPPP workshop - BMVC*, page 13. British Machine Vision Association, 2015.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[13] K. He, X. Zhang, S. Ren, and J. Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, pages 346–361. Springer International Publishing, Cham, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun. 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.

[16] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[17] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR0*, 2015.

[18] M. Koornneef, C. Hanhart, P. van Loenen-Martinet, and H. Blankestijn de Vries. The effect of daylength on the transition to flowering in phytochrome-deficient, late-flowering and double mutants of arabidopsis thaliana. *Physiologia Plantarum*, 95(2):260–266, 1995.

[19] V. Lempitsky and A. Zisserman. Learning To Count Objects in Images. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1324–1332. Curran Associates, Inc., 2010.

[20] G. Lobet. Image Analysis in Plant Sciences: Publish Then Perish. *Trends in Plant Science*, may 2017.

[21] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters*, 81:80–89, 2016.

[22] M. Minervini, H. Scharr, and S. A. Tsaftaris. Image Analysis: The New Bottleneck in Plant Phenotyping. *IEEE Signal Processing Magazine*, 32(4):126–131, jul 2015.

[23] J.-M. Pape and C. Klukas. Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images. *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)*, pages 1–12, 2015.

[24] M. Ren and R. S. Zemel. End-to-End Instance Segmentation and Counting with Recurrent Attention. 2016.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[26] B. Romera-Paredes and P. H. S. Torr. Recurrent Instance Segmentation. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, pages 312–329. Springer International Publishing, Cham, 2016.

[27] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, X. Yin, and S. A. Tsaftaris. Leaf segmentation in plant phenotyping: a collation study. *Machine Vision and Applications*, 27(4):585–606, may 2016.

[28] A. Telfer, K. Bollman, and R. Poethig. Phase change and the regulation of trichome distribution in arabidopsis thaliana. *Development*, 124(3):645–654, 1997.

[29] S. A. Tsaftaris, M. Minervini, and H. Scharr. Machine Learning for Plant Phenotyping Needs Image Processing. *Trends in Plant Science*, xx:1–3, 2016.

[30] A. Walter and U. Schurr. The modular character of growth in nicotiana tabacum plants under steady-state nutrition. *Journal of Experimental Botany*, 50(336):1169, 1999.

[31] A. Yao, J. Gall, C. Leistner, and L. V. Gool. Interactive object detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3242–3249, June 2012.

[32] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *Computer Vision ECCV 2014*, 8689:818–833, 2014.