

# Deep Learning for plant identification using vein morphological patterns

Guillermo L. Grinblat<sup>1</sup>, Lucas C. Uzal<sup>1</sup>, Mónica G. Larese<sup>1</sup> and Pablo M. Granitto<sup>1</sup>

<sup>1</sup>CIFASIS, French Argentine International Center for Information and Systems Sciences, UNR-CONICET, Argentina

May 27, 2016

## Abstract

We propose using a deep convolutional neural network (CNN) for the problem of plant identification from leaf vein patterns. In particular we consider classifying three different legume species: white bean, red bean and soybean. The introduction of a CNN avoids using handcrafted feature extractors as in state of the art pipeline. Furthermore, this deep learning approach significantly improves the accuracy of the referred pipeline. We also show that this accuracy is reached by increasing the depth of the model. Finally, by analyzing the resulting models with a simple visualization technique, we are able to discover which vein patterns are relevant.

**Keywords:** Deep Learning, Machine Vision, Automatic Plant Identification, Leaf Vein Image.

## 1 Introduction

Nowadays, in many typical applications of machine vision there is a tendency to replace classical techniques with deep learning algorithms [1]. The benefits are valuable: on one hand it avoids the need of specialized handcrafted features extractors and, on the other hand, results are not damaged, moreover they typically get improved. Some examples of this can be found in [2–4].

Deep learning refers to training neural network architectures composed of several nonlinear processing layers. The success of deep learning is based on new model regularization techniques [5], improved nonlinearities design [6], and current hard-

ware capabilities, among others. In particular, for Machine Vision tasks, the success of deep learning is based on convolutional neural networks (CNN, [7]) which have become the standard neural network variant to process images [1].

There are many agricultural problems currently addressed with classical machine vision techniques that may benefit from using a deep learning approach. We consider in this paper a successful example of this behavior by applying deep learning to automatic plant identification.

Automatic plant identification is a challenging problem that has drawn the attention of many works in the recent years, in particular those based on leaf image analysis. Much of this work make use leaf features that humans can perceive. The goal of automatization in this case is to avoid human experts to handle huge catalogs of plant species and to reduce classification time. Some works are focused on leaf shape [8–14], some use shape and texture [15], while others consider color and texture [16].

Recently, however, more attention has been payed to vein morphological patterns as a fingerprint of the leaf. A clear correlation has been established between vein characteristics and some properties of the leaf (such as damage and drought tolerance, among others) [17, 18]. This suggests that vein morphology carries information suitable for plant classification when shape, color or texture differences are not observable, as in the case of trying to separate different cultivars from the same species. This kind of features may not be easily spotted by a human observer and automated

recognition becomes indispensable.

Following this premise, Larese et al. [19] performed computer vision techniques to extract several vein morphological measures and showed that it is possible to separate three different plant species by using only this information and supervised machine learning algorithms. In a later work [20], they used similar techniques to reach some degree of discrimination between plants belonging to different cultivars of the same species.

In this work we evaluate a deep learning model for this predefined problem of plant identification based on vein morphology. We show that the plain application of such learning model leads us to better results than those obtained with a standard machine vision pipeline. Furthermore, the utilization of a simple model visualization technique allows us to identify meaningful vein patterns. All these successful results on plant classification from leaf vein morphology are a valuable step for motivating further research and extension of our leaf database with more species and cultivars.

The rest of the paper is organized as follows. In Sec. 2 we review the task specific approach for the problem at hand as proposed by Larese et al. [20]. In Sec. 3 we introduce the proposed deep learning approach and explain the performed experiments in Sec. 4. Results are presented in Sec. 5 and in Sec. 6 we show which patterns resulted relevant for classification. Finally, we draw some conclusions in Sec. 8. In Appendix A a more detailed information about the data acquisition and processing can be found.

## 2 Task specific approach

Many successful Machine Learning applications make intensive use of specific knowledge about the task provided by human experts. In this section we summarize the approach considered in [20] for plant classification based on leaf veins, which makes use of expert knowledge.

The processing pipeline is divided in four stages as shown in Fig. 1. The starting point is the set of images of first foliage leaves acquired with a standard flatbed scanner (see Appendix A for more details). These images are processed according to the following stages:

(i) Vein Segmentation. First, an unconstrained

version of the Hit or Miss Transform (UHMT) [21] is applied, in order to extract the vein morphological patterns. The output of this transform is a binary image and therefore it eliminates color information.

- (ii) Central patch extraction. A central patch (100x100 pixels) of the binary image is cropped and the rest of the image is discarded, in order to dismiss any influence of the leaf shape.
- (iii) Vein measures. At this stage a set of features of interest was extracted with the help of LEAF GUI [22]. This set includes measures such as total number of veins, total number of nodes, mean vein width, among others.
- (iv) Classification. Three different Machine Learning algorithms were tested: Support Vector Machines (SVM), Penalized Discriminant Analysis (PDA) and Random Forests (RF) [23]. These models were trained using the features obtained in the last step.

There are two main observations about this pipeline. First, in order to highlight different levels of vein details, Larese et al. applied the UHMT to resized versions of the leaf image. The scale factors considered were 100% (no resize), 80%, and 60%. The processed images were resized back to the original size. With these three output images, two alternatives were studied. In the first one, a single combined image was obtained by adding them. For the second alternative, the three output images and the combined one are kept. We will refer to this two setups as S1 and S2 respectively. Figure 2 shows some example images after stage (ii), for the S2 setup. The S1 setup correspond to select only the first column for each sample.

The second point to notice is that stage (iii) is the only one that require specific domain knowledge. All the considered measures can be automatically extracted but were designed by experts specifically to characterise vein patterns.

Also, it is important to remark that the number of features extracted in stage (iii). Larese et al. extracted 52 features from each patch image. This means 52 features in the S1 setup, but 208 features in the S2 setup.

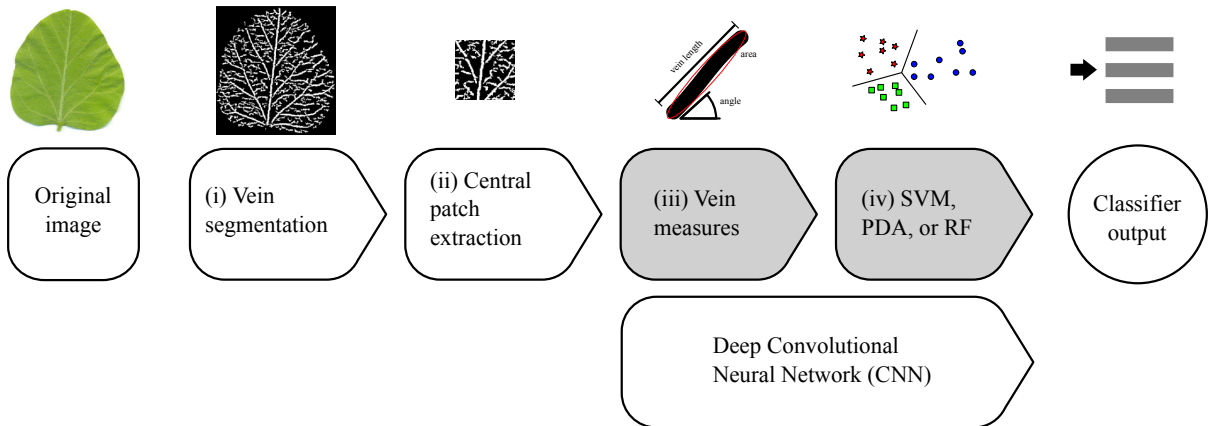


Figure 1: Adopted pipeline (as in Larese et al. [20]). In this work the greyed stages were replaced by a deep convolutional network. Stages (i) and (ii) were kept in order to allow a fair comparison with Larese et al. results. By design, this two stages filter color and leaf shape information.

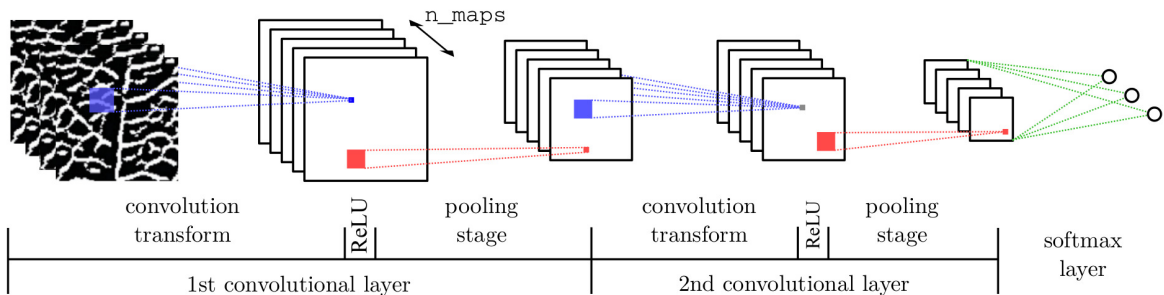


Figure 3: Diagram of a three layer version of the CNN considered in this work. The first two are convolutional layers and the last one is a softmax layer. We evaluate CNNs up to 6 layers.

### 3 Deep Learning proposal

In this work we replace stages (iii) and (iv) of the previous pipeline (see Fig. 1) with a CNN (explained in the next subsection). We expect that the CNN can automatically learn from the training set the appropriate features to solve the classification problem. This implies that we can put aside the expert knowledge for designing the features.

It should be noticed that stages (i) and (ii) ensure that color and leaf shape information is deleted. The reason for this is that the main objective in [20] is to show that it is possible to classify plant species only from vein morphological patterns. It should be remarked that this setup mimics the case where it is expected to find only morphological differences, as in the case of different cultivars from the same species.

In order to fairly compare our results with those from [20], we also include stages (i) and (ii) in our pipeline. That is, we consider the classification problem using only the vein morphological patterns, passing to the CNN images as in Fig. 2.

#### 3.1 Convolutional Neural Networks

CNNs were first introduced by Y. LeCun [7]. They have an architecture specially designed to deal with images. In this section we make a brief introduction to CNNs mainly for defining the hyperparameters considered in this work. For a more detailed description of this kind of models we refer the reader to [24] and the references therein.

Figure 3 depicts a diagram of the considered CNN model. Each layer is composed of three transforms. First, there is a convolution operation be-

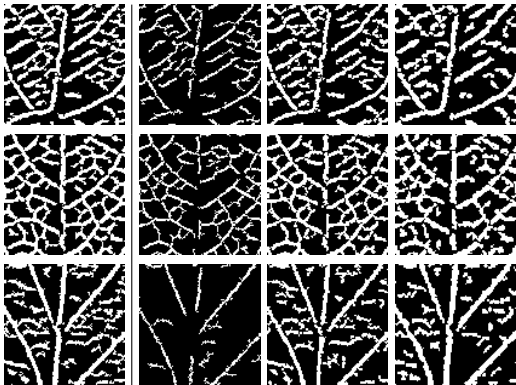


Figure 2: Image samples obtained after processing stages (i) and (ii). The first column correspond to preprocessing S1, while setup S2 is formed by all the columns. These images are the input to the CNN.

tween the input image and a filter bank of size `n_maps`. Each filter has a bounded size associated to a small receptive field in the input image (blue squares in Fig. 3). We use square filters of width `filter_size`. For each filter in the bank the convolution produces a feature map.

The second transform is an elementwise nonlinear function applied to all the feature maps. Typically this function is a ReLU<sup>1</sup>.

Finally, there is a subsampling transform. In this pooling stage, each map is divided in a set of non-overlapping square neighborhoods of width `pool_size` (red squares). From each neighborhood, this transform only retains the maximum value.

The last layer in the network is a softmax function. It returns the estimated probability of each class, given a concrete sample. This layer is fully connected to all the output feature maps of the last convolutional layer.

The described topology produces a huge reduction in the number of free trainable parameters in comparison to a standard (fully connected) artificial neural network. This is due to the sparse neural connectivity (restricted to small receptive fields) and to the sharing of filter values along image locations exploiting translational invariance.

A last relevant comment on CNNs is about vi-

<sup>1</sup> $ReLU(x) = \max(0, x)$

sualization techniques. Recently published procedures [25] allow visualizing which patterns are detected at each layer of the deep network, partially removing the CNNs from the category of black-box models. In this paper we use a simple procedure considered in [25] for highlighting the most relevant input image regions for the network output probabilities (see Sec. 6).

## 4 Experiments

We evaluated the proposed pipeline with a CNN stage by performing a set of numerical experiments in which several models were trained in order to estimate test error. Below we detail the experiment outlines.

**Datasets.** We consider the leaf images already processed with stages (i) and (ii) of the original pipeline (Fig. 1) with the two setup variations S1 and S2 (as explained in Sec. 2) as input of our CNN without any further processing. For the S2 setup, in which we have four images per sample, they are considered as four different input channels for the CNN (as it is standard with the three channels in a RGB image). We have therefore two series of experiments associated with these two setups S1 and S2, as in [20]. Summarizing, the input of the CNN is a 100x100x1 binary image in the setup S1 and a 100x100x4 one in the setup S2.

**Models.** We trained CNN models of increasing depth: from 2 layers (1 conv. layer + 1 softmax) to 6 layers (5 conv. layers + 1 softmax). The architecture is the same for all the convolutional layers in each model. That is fixed `filter_size` and `n_maps` for the convolution transform followed by a 2x2 pooling and a ReLU elementwise nonlinearity<sup>2</sup>.

**Training algorithm.** The parameters were optimized using stochastic gradient descent (SGD) over a training set using minibatches of 20 samples. We use a 50% dropout rate [5] in the training stage for regularization. After some preliminary training experiments we set the learning rate at 0.01 and set momentum to zero.

<sup>2</sup>The best model configuration obtained by cross validation for a 5-layer CNN has `filter_size=12` and `n_maps=10`. See Sec. 5.

**Error estimation.** For each model depth, the final test errors were estimated, as in [20], averaging over 100 runs (10 independent runs of 10-fold cross-validation). The hyperparameters `filter_size` and `n_maps` were chosen by minimizing the classification error over a validation set. We used early stopping for the SGD iterations by monitoring the validation error.

**Implementation.** All the experiments were carried out using Pylearn2 [26].

## 5 Results

Figures 4 and 5 show that the final accuracy consistently improves with the depth of the model, with independence of the selected setup (S1 or S2).

For S1 setup (Fig. 4) the best performance is reached at a depth of 5 layers, attaining a mean accuracy of  $(92.6 \pm 0.2)\%$ . This value surpasses the best mean accuracy reported in [20] corresponding to a PDA classifier ( $90.4 \pm 0.3$ , horizontal dashed line in Fig. 4). The reached accuracy does not improve by adding extra layers: there is no clear difference between 5 and 6 layer results.

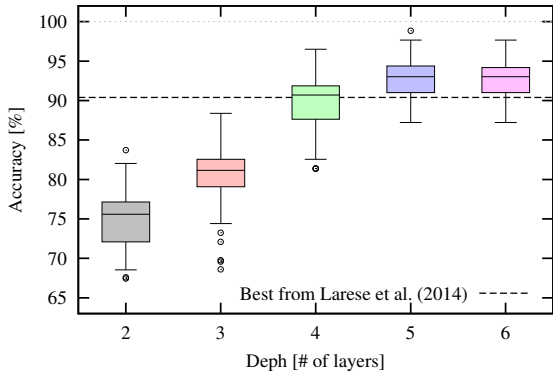


Figure 4: Classification accuracy as a function of CNN depth for setup S1. Each box resumes the results from 100 runs (10 runs of 10-fold cross validation) for the corresponding depth. The box size covers the two central quartiles, while the whiskers span the central 95% of the runs. The horizontal dashed line indicates the best mean value obtained by a task specific approach [20] for this setup.

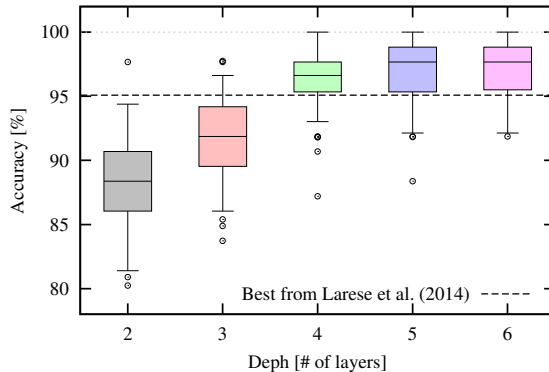


Figure 5: Classification accuracy as a function of CNN depth for setup S2. See Fig. 4 caption for details.

For the S2 setup (Fig. 5) we find similar results. The best performance is also reached with the 5-layer model with no clear difference with the 6-layer results. As in [20] the accuracy levels with setup S2 are consistently higher than with setup S1. In this case the 5-layer model accuracy reaches a mean value of  $(96.9 \pm 0.2)\%$ , surpassing the  $(95.1 \pm 0.2)\%$  best accuracy in [20].

Table 1 shows the mean accuracy for each class for the best models. The improved performance of the 5-layer model in both setups, S1 and S2, comes mainly from a better classification of red bean leaves.

For the S1 setup the accuracy of the 5-layer model over white bean leaves drops significantly with respect to the best model of [20]. However, as white bean is the least numerous class, this drop is not enough for changing the overall result. This behavior banishes for the S2 setup, where the accuracy of the 5-layer model is similar or better for all classes.

Finally, we show Receiver Operating Characteristic (ROC) curves for the S2 setup in order to evaluate classifier output quality. Figure 6 show one-vs-all ROC curves for each class for the different models (from 2 to 6 layers). Consistently, white beans are harder to classify for all depths. However, we found that all curves improves with depth until layer 5 model. Bottom left panel shows average curves for different depth models where the 6-layer model fall below the 5-layer model curve.

Model	White bean (%)	Red bean (%)	Soybean (%)	Total (%)
<i>Combined veins only (S1 setup):</i>				
PDA, best of [20]	82.7 $\pm$ 0.9	85.8 $\pm$ 0.6	96.5 $\pm$ 0.3	90.4 $\pm$ 0.3
CNN 5 layers	77.6 $\pm$ 1.1	93.8 $\pm$ 0.5	98.8 $\pm$ 0.2	93.0 $\pm$ 0.3
<i>Combined veins with 3 scales (S2 setup):</i>				
PDA, best of [20]	90.9 $\pm$ 0.6	91.7 $\pm$ 0.5	99.0 $\pm$ 0.1	95.1 $\pm$ 0.2
CNN 5 layers	90.2 $\pm$ 1.0	98.3 $\pm$ 0.3	98.8 $\pm$ 0.2	96.9 $\pm$ 0.2

Table 1: Per class and total accuracy (mean  $\pm$  stdev). Values correspond to CNN model and best accuracies reached in [20] for the S1 and S2 setups.

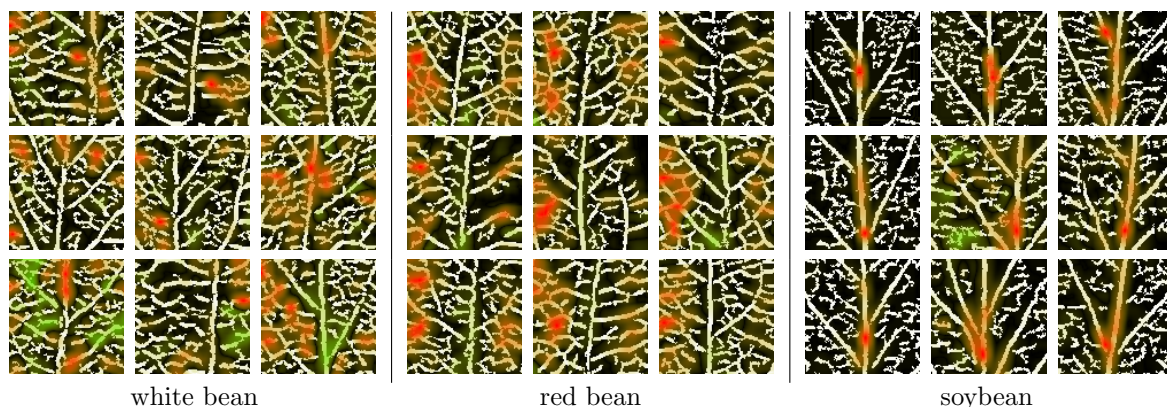


Figure 7: Heatmaps indicating the locations of the input images to which the network correct-class output probability is more sensitive to partial occlusion. Red colored regions correspond to a decrease of the output probability while green indicates an increase. Each panel shows 9 examples from each class.

## 6 Visualizing relevant patterns

In order to have an idea of which patterns an already trained CNN is attending for class labeling we perform an experiment similar as in [25]. We occlude different parts of an input image with a 10x10 black patch and we observe the variation in the output probability for the correct class<sup>3</sup>. The result is a heatmap indicating the locations of the image to which the output is more sensitive. Figure 7 shows some example images for the three classes. Red colored regions correspond to a decrease of the output probability while green indicates an increase. As all the shown examples were correctly classified, this occlusion procedure typically lowers the probability

<sup>3</sup>We use a trained 5-layer CNN with `filter_size=12` and `n_maps=10` in S2 setup.

of the correct class, biasing the heatmap results to red as observed. We can see that different parts of the vein patterns are taken into account for each class. For example, in the case of the soybean samples, the most sensitive regions are in the central vein and in particular, in the place where the secondary veins join to it. However, for white and red bean classes, the results suggest that the outer and smaller veins are also relevant for the network output. In these highlighted regions we expect to find the patterns relevant for the task at hand.

Some selected samples shown in Fig. 8 help us to understand the tradeoff between the output probabilities for the three classes. Rows (a) and (b) show two samples images from white bean leaves, while rows (c) and (d) correspond to a single sample image from red bean and soybean respectively. Each column shows the variation of the output probabilities of the model associated to each class, when

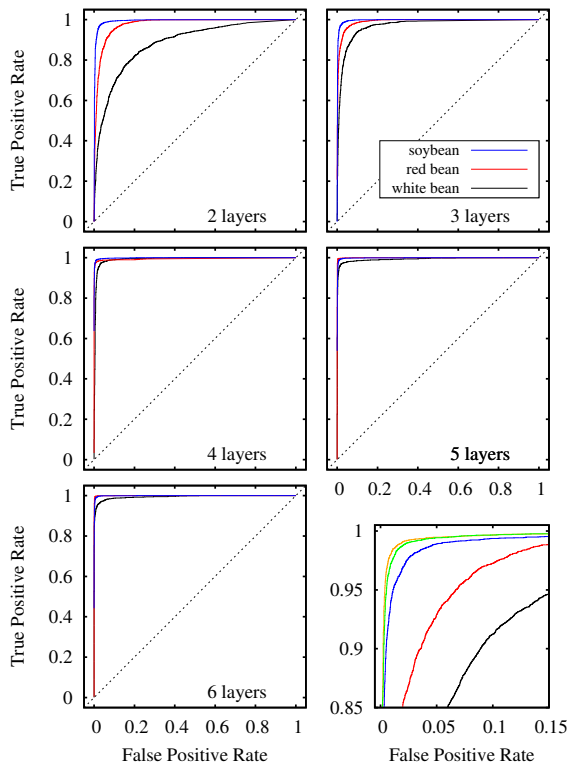


Figure 6: Receiver Operating Characteristic (ROC) curves for the S2 setup. The first 5 panels show one-vs-all ROC curves for each class for the different models (from 2 to 6 layers). Bottom left panel shows average curves for different depth (black, 2 layers; red, 3 layers; blue, 4 layers; orange, 5 layers; and green, 6 layers).

performing the same procedure as in Fig. 7. Given that these probabilities sum 1, a decrease in one class (red regions) is associated to an increase in the others (green regions).

The first thing we can observe from rows (a) and (b) is that by occluding the higher order veins in the left side of the image the correct class output probability drops (first column) in favor of soybean probability (third column). This is because soybean leaves' higher order veins are sparser than in the case of red and white beans (see samples in Fig. 7). Another observation we can make concerns the central vein in row (b). This particular white bean presents these lower order veins similar to a soybean, given its ramification angle. This

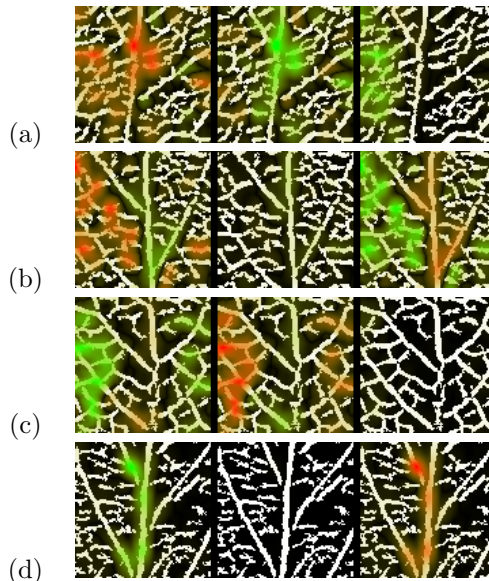


Figure 8: Heatmaps indicating the locations of the input images to which each class output probability is more sensitive to partial occlusion. The columns correspond to each of the three outputs (white bean, red bean, and soybean). Red colored regions indicate a decrease of the corresponding output probability while green, an increase. Rows (a) and (b) correspond to white bean images; (c), to red bean; and (d), to soybean. For these examples, the variation of the probability assigned to the correct class is also shown in Fig. 7.

is why when we occlude parts of the central vein the probability of the correct class increases, at the expense of the soybean probability.

Regarding row (c), the selected red bean sample shows a pattern typically found in that class. The cell-like pattern in the higher order veins seems to be a subtle difference with the white bean class. Therefore, when occluding these patterns the probability of red bean decrease (second column) in favor of the white beans.

Rows (c) and (d) also show a typical behavior of the red bean and soybean classes. Both classes interact with the white bean class in the probability exchange but they do not interact with each other. This suggests that these two classes do not have a frontier in common in the representation computed by the deep network at the input to the softmax

layer.

## 7 Discussion: overall efficiency

When comparing two approaches for solving a specific task it is advisable to consider the overall efficiency, that is the ratio between the results quality measure and the weight of the resources employed. In Sec. 5 we present a comparative summary in terms of classification accuracy. The counterpart of resources spent is harder to compare given that it requires to measure time and effort spent designing and training neural models vs. manually designing and selecting feature extractors and performing classification. However, we can state a few points for helping comparison.

Designing features extractors is labor-intensive and require expert skills. We have no available data about resources employed in ref. [22] for designing leaf vein features. Nevertheless, for most vision applications, suitable feature extractors can be found publicly available as in this case. The effort is transferred however to the search and selection of proper feature extractors for the task, within a very large corpus of machine vision literature. This learning process is the most time consuming task and can take from one to two weeks of research. Once selected features, performing classification can be done with a predefined application agnostic pipeline. The evaluation of different classification algorithms (SVM, PDA, RF) and cross validation selection of hyperparameters can be automated and requires few hours of computational load on a standard CPU.

On the other hand, for our proposal, we just use a general purpose standard CNN configuration (convolutional layer + ReLU activation + dropout, [1]). An implementation of such network can be found as a toy example in any deep learning library. We add none special feature to adapt this standard model to this particular application. Therefore, having the model trained on the leaf vein dataset with competitive validation accuracies take us a few hours. The rest of the time (around two weeks<sup>4</sup>) was spent designing and running cross validation experiments (as in any classification task) for pa-

<sup>4</sup>Repeating single training 50 times

rameter optimization and error estimation which is only hardware intensive.

Summarizing, we cannot precisely state which pipeline is less time consuming. However, deep learning approach relieves research for feature extractors by automatic learning proper features. This conveniently transfers human expert time to computational load and, for this specific application, we also get improved accuracies.

## 8 Conclusion

In this work we show a successful example of application of deep learning in the area of agriculture, specifically plant identification from leaf vein patterns. We replaced a task specific module in a state of the art processing pipeline with a deep convolutional network. The main result is that we obtained an improved accuracy using a standard deep learning model. This implies that it is not necessary to handcraft a specific feature extraction method for this task. We also showed that the accuracy monotonically improves with the depth of the model. This implies that depth is a key element for solving the problem.

It is commonly argued that neural network solutions does not shed light on the solved problems as they are black box models. However, using a simple visualization technique, we obtained the relevant vein patterns for the classification task performed by the deep model.

We are currently exploring other potential applications of deep learning in agriculture, as weeds detection and identification or seeds viability tests.

## Acknowledgments

Authors acknowledge grant support from ANPCyT PICT-2012-0181.

## A Dataset

In this work we use the dataset introduced by [19]. It is formed by 866 leaf images provided by INTA (Instituto Nacional de Tecnología Agropecuaria, Oliveros, Argentina). It is divided in three classes: 422 images correspond to soybean leaves, 272 to red bean leaves and 172 to white bean leaves. These are



first foliage leaves after 12 days of seedling grow. The images were acquired using a standard flatbed scanner (Hewlett Packard Scanjet-G 3110) at a resolution of 200 pixels per inch. The images correspond to the abaxial surface of the leaves.

## B Processing pipeline details

In this appendix we describe in more detail the first two stages in the processing pipeline. That is, the stages that we maintain in the proposed method.

- (i) The color information was removed by converting the RGB images to grayscale.
- (ii) A binary mask was obtained for each leaf, using the automatic iterative threshold selection algorithm [27]. Its holes were filled using morphological reconstruction [21] and finally all the connected components except the largest one were removed.
- (iii) An unconstrained version of the Hit or Miss Transform (UHMT) [21] on five different sized (100%, 90%, 80%, 70% and 60%) versions of the images were computed. For this purpose, four composite structuring elements were used aimed at detecting leaf veins in four directions: vertical, horizontal,  $+45^\circ$  and  $-45^\circ$  (shown in Fig. 9).
- (iv) The resulting UHMTs were resized back to its original size and added to obtain the combined UHMT. The resized UHMTs at 100%, 80% and 60% were also preserved.
- (v) An adaptive histogram equalization and adaptive thresholding was performed and all the connected components with less than 20 pixels were removed.
- (vi) The UHMTs were masked using the result of step (ii).
- (vii) Finally, a central patch (100x100 pixels) of the binary image is cropped and the rest of the image is discarded. Since the images are taken with a resolution of 200 dpi, this corresponds to a square of 0.5x0.5 inches roughly at the center of the leaf.

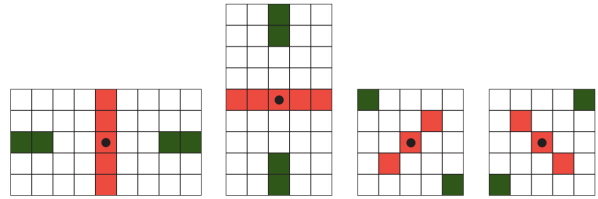


Figure 9: The four pairs of flat composite structuring elements used in the UHMT computation to detect veins in four directions, from left to right: vertical, horizontal,  $+45^\circ$ , and  $-45^\circ$ . Foreground pixel configurations are depicted in red while background pixel configurations are in green. The center of the composite structuring element is marked with a black dot.

## References

- [1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [3] Dan C Cireřan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, pages 411–418. Springer, 2013.
- [4] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lars Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014.
- [5] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

- [6] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE, 2013.
- [7] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.
- [8] G. Agarwal, H. Ling, D. Jacobs, S. Shirdhonkar, W. Kress, R. Russell, P. Belhumeur, N. Dixit, S. Feiner, D. Mahajan, K. Sunkavalli, , and S. White. First steps toward an electronic field guide for plants. *Taxon, Journal of the International Association for Plant Taxonomy*, 55:597–610, 2006.
- [9] J. Camargo Neto, G. E. Meyer, D. D. Jones, and A. K. Samal. Plant species identification using Elliptic Fourier leaf shape analysis. *Computers and Electronics in Agriculture*, 50:121–134, 2006.
- [10] Jyotismita Chaki and Ranjan Parekh. Designing an automated system for plant leaf recognition. *International Journal of Advances in Engineering & Technology*, 2(1):149–158, 2012.
- [11] Ji-Xiang Du, Xiao-Feng Wang, and Guo-Jun Zhang. Leaf shape based plant species recognition. *Applied Mathematics and Computation*, 185(2):883–893, 2007. Special Issue on Intelligent Computing Theory and Methodology.
- [12] Chih-Ying Gwo, Chia-Hung Wei, and Yue Li. Rotary matching of edge features for leaf recognition. *Computers and electronics in agriculture*, 91:124–134, 2013.
- [13] Cholhong Im, Hirobumi Nishida, and Toshiyasu L Kunii. Recognizing plant species by leaf shapes—a case study of the Acer family. *Pattern Recognition, International Conference on*, 2:1171, 1998.
- [14] Jordi Solé-Casals, Carlos M. Travieso, Jesús B. Alonso, and Miguel A. Ferrer. Improving a leaves automatic recognition process using PCA. In *IWPACBB*, pages 243–251, 2008.
- [15] Z Husin, AYM Shakaff, AHA Aziz, RSM Farook, MN Jaafar, U Hashim, and A Harun. Embedded portable device for herb leaves recognition using image processing techniques and neural network algorithm. *Computers and Electronics in agriculture*, 89:18–29, 2012.
- [16] R. Pydipati, T. F. Burks, and W. S. Lee. Identification of citrus disease using color texture features and discriminant analysis. *Computers and Electronics in Agriculture*, 52:49–59, 2006.
- [17] L. Sack, E. M. Dietrich, C. M. Streeter, D. Sanchez-Gomez, and N. M. Holbrook. Leaf palmate venation and vascular redundancy confer tolerance of hydraulic disruption. *PNAS USA*, 105:1567–1572, 2008.
- [18] C. Scoffoni, M. Rawls, A. D. McKown, H. Cochard, and L. Sack. Decline of leaf hydraulic conductance with dehydration: relationship to leaf size and venation architecture. *Plant Physiology*, 156:832–843, 2011.
- [19] Mónica G Larese, Rafael Namías, Roque M Craviotto, Miriam R Arango, Carina Gallo, and Pablo M Granitto. Automatic classification of legumes using leaf vein image features. *Pattern Recognition*, 47(1):158–168, 2014.
- [20] Mónica G Larese, Ariel E Bayá, Roque M Craviotto, Miriam R Arango, Carina Gallo, and Pablo M Granitto. Multiscale recognition of legume varieties based on leaf venation images. *Expert Systems with Applications*, 41(10):4638–4647, 2014.
- [21] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, 1999.
- [22] Charles A Price, Olga Symonova, Yuriy Mileyko, Troy Hilley, and Joshua S Weitz. Leaf extraction and analysis framework graphical user interface: segmenting and analyzing the structure of leaf veins and areoles. *Plant Physiology*, 155(1):236–245, 2011.
- [23] T. Hastie, Tibshirani R., and J. Friedman. *The Elements of Statistical Learning, Second Edition*. Springer, 2009.

- [24] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.
- [25] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.
- [26] Ian J. Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*, 2013.
- [27] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.