Artificial Intelligence Research and Development
M. Villaret et al. (Eds.)

doi:10.3233/FAIA210157

Reliable Deep Learning Plant Leaf Disease Classification Based on Light-Chroma Separated Branches

Joao Paulo SCHWARZ SCHULER [a], Santiago ROMANI [a]
Mohamed ABDEL-NASSER [a] Hatem RASHWAN [a] and Domenec PUIG [a]
[a] *Universitat Rovira i Virgili*

Abstract. The Food and Agriculture Organization (FAO) estimated that plant diseases cost the world economy $220 billion in 2019. In this paper, we propose a lightweight Deep Convolutional Neural Network (DCNN) for automatic and reliable plant leaf diseases classification. The proposed method starts by converting input images of plant leaves from RGB to CIE LAB coordinates. Then, L and AB channels go into separate branches along with the first three layers of a modified Inception V3 architecture. This approach saves from 1/3 to 1/2 of the parameters in the separated branches. It also provides better classification reliability when perturbing the original RGB images with several types of noise (salt and pepper, blurring, motion blurring and occlusions). These types of noise simulate common image variability found in the natural environment. We hypothesize that the filters in the AB branch provide better resistance to these types of variability due to their relatively low frequency in the image-space domain.

Keywords. DCNN, CNN, Plant Leaf Disease, Classification, Computer Vision, Plant Village, Deep Learning

1. Introduction

Plant leaf images taken in the field and away from controlled laboratory conditions frequently suffer from blurring, motion blurring, occlusion and illumination variations. Automated detection systems frequently suffer from these common adverse effects. Inspired on Multi-path Convolutional Neural Networks [1] and Dual Paths Neural Networks [2], we created an Inception V3 [3] based architecture that has two branches (paths) along the first 3 convolutional layers. One branch is fed with the achromatic L channel, while the other branch is fed with AB channels provided by the input CIE Lab color coordinate space. In this work, we study 3 two-branches Inception V3 variants: 20%L-80%AB, 50%L-50%AB and 80%L-20%AB. In this notation, the percentages indicate the proportion of the original number of neurons of each separated layer dedicated to each path. This two-branches solution provides more resistance to adverse effects such as blurring. For this work, we are training our architecture with the PlantVillage dataset [4] that contains classes for 12 healthy crops and 26 crop diseases.

This article is structured as follows: section 2 presents and discusses relevant work in regards to computer vision, DCNNs and image based plant disease diagnostic. Section 3 presents the proposed method. The results and the discussion are given in sections 4 and 5. Section 6 summarizes the main conclusions.

2. Related work

In a previous work [5], training a CNN with input images encoded in the CIE Lab color space, we were able to show that we can classify the CIFAR-10 dataset [6] more efficiently and with higher classification accuracy by creating an architecture that has a subpath dedicated to light and another subpath dedicated to color channels. In this previous work, each subpath has only the first convolution layer dedicated for each L and AB channels.

A number of machine learning methods have been proposed specifically for image based plant disease diagnostic [7,8]. Mohanty et. al. [9] worked with AlexNet and GoogLeNet models for the PlantVillage dataset classification. They trained both models from scratch and with transfer learning. They also experimented feeding their models with RGB and grayscale images. They found better results feeding RGB images to both tested models. Their best result without transfer learning was 98.37%. Geetharamani et al. [10] classified the PlantVillage dataset with 3 convolutional, 2 max poolings and 2 dense layers achieving 96.46% of accuracy. Toda at al. [11] working with a trimmed Inception V3 showed that DCNNs can learn the colors and textures specific to plant leaf diseases resembling human made classification.

3. Methodology

Figure 1 shows two designs of CNNs for plant disease classification. Toda & Okura's [11] proposed an Inception v3 variation that gets rid of the last 5 mixed layers (out of 11). The authors proved that it is enough for the sake of classification PlantVillage dataset. Therefore, we have chosen their model as our baseline.

The design shown on the right of figure 1 corresponds to our proposal, which splits the first three convolution layers of the baseline model into two branches, one for the L channel and the other for the AB channels from the transformed RGB image. Then, the output from each branch is concatenated and the rest of the network is the same as the baseline.

Another relevant remark is that we use a hyperparameter that determines the distribution of a fixed number of filters among L and AB branches, which allows us to look for the optimal contribution of each branch to the classification task. This distribution is implemented with the value of a variable x, shown in figure 1 as the number of L filters in the third layer. In the original Inception V3 implementation, the first three convolutional layers have 32, 32 and 64 filters, respectively. We have analyzed three configurations of the two-branch design named after the percentage of filters dedicated to L and AB branches: 20%L-80%AB, 50%L-50%AB and 80%L-20%AB. The resulting number of filters for each variant is shown in the table 1.

Since we intend to compare our variants with the baseline as fairly as possible, the sum of filters of the two branches in each layer is the same as in the Inception V3 design.

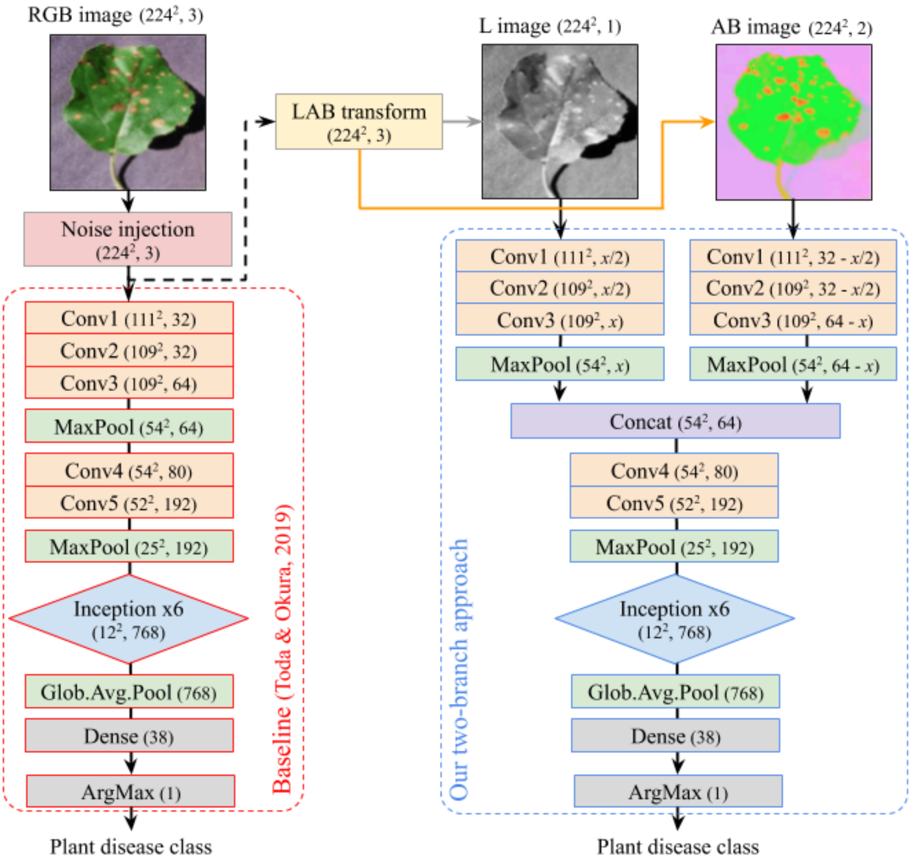

Figure 1. Graphical representation of the worked network architectures: on the left, the Toda & Okura's single-branch (baseline) approach fed with an RGB image; on the right, our two-branch approach fed with L+AB images. The x expressions determine a varying number of filters in L branch and a complementary number of filters in AB branch.

Model	1st & 2nd Layers	3rd Layer
baseline	32	64
20%L + 80%AB	6 — 26	13 — 51
50%L + 50%AB	16 — 16	32 — 32
80%L + 20%AB	26 — 6	51 — 13

Table 1. Number of filters in 1st, 2nd and 3rd layers of the baseline and our variants. For our variants, we have the number of filters in the L branch at the left and in the AB branch at the right.

However, our design saves from 1/3 to 1/2 of weights and computational floating point operations in the split layers, as shown in tables 2. Despite the reduction in weights, the learning capacity of our models is not degraded since our three variants achieve similar accuracy (99.48%, 99.11%, 99.08%) to the one provided by the baseline (99.32%).

Our design is based on the well-known fact that RGB channels are highly correlated among each other [12] in the sense that shading and shadows render a set of different RGB values from the intrinsic color(s) of a surface. Specifically, intensity variations induced by illumination variation, edges and texture modify the three RGB values at same proportion. Hence, transforming RGB channels into some sort of achromatic-chromatic space, like CIE Lab, effectively isolates the gray-level features in the L channel and the color-related features in the AB channels. We are forcing the filters in each branch to learn features related to the nature of each cue, i.e., we expect that L filters will focus on texture and edges of the leafs (intrinsic shape, damaged leaf areas, etc.) while the AB filters will focus on color findings (lesions, general color of the leaf, etc.).

model	weights (Saving)	flops (Saving)
baseline	28512	701M
20%L + 80%AB	19746 (31%)	485M (31%)
50%L + 50%AB	14256 (50%)	350M (50%)
80%L + 20%AB	19566 (31%)	481M (31%)

Table 2. Weights and required forward pass floating point operations along the first 3 convolutional layers in baseline and our variants.

To verify the reliability of the baseline and our variants, we have included one module for noise injection. This allows us to perturb the original RGB images with different types of artifacts and varying degrees of severity of those artifacts. It must be observed that the noise injection is previous to the RGB-to-LAB transformation.

Our code was coded with Keras/Tensorflow v2.2. We rented cloud based hardware with NVIDIA GPUs, intel CPUs and virtual machines from 32GB to 64GB of RAM. The implementation details of our approach are strongly based on the reference paper [11]. Each convolutional layer is composed of a 2D convolution, a batch normalization and a ReLU activation function. All convolutional filters from Conv1 to Conv5 are of the size 3×3 except for Conv4 which is 1×1. The optimization method is stochastic gradient descent, and the loss function is weighted categorical cross entropy to compensate for unbalanced number of samples among classes. The batch size is 32 and we store the weights that obtain the best validation accuracy in 30 epochs. We trained all models from scratch. The noise injection module has not been used for training since this module is only intended to verify the reliability of the models under controlled perturbation of the test images.

4. Results

Figure 2 shows the evolution of test accuracy in the studied models, baseline, two-branch 20%L-80%AB, 50%L-50%AB and 80%L-20%AB, for different types of noise and a range of noise amount.

In Salt and Pepper experiments, the range of noise indicates the percentage of pixels of the input image that have been changed to either white or black pixels (see Fig. 3 for an example). This type of noise simulates spuriously saturated values in the input signal. The corresponding plot depicts the 20%L-80%AB variant as the most reliable when the percentage of noisy pixels is above 3%. Above 3%, the classification accuracy is up to 10% more accurate than the baseline. Nevertheless, the baseline holds better performance than the other two branched models in the range of noise used for these experiments.

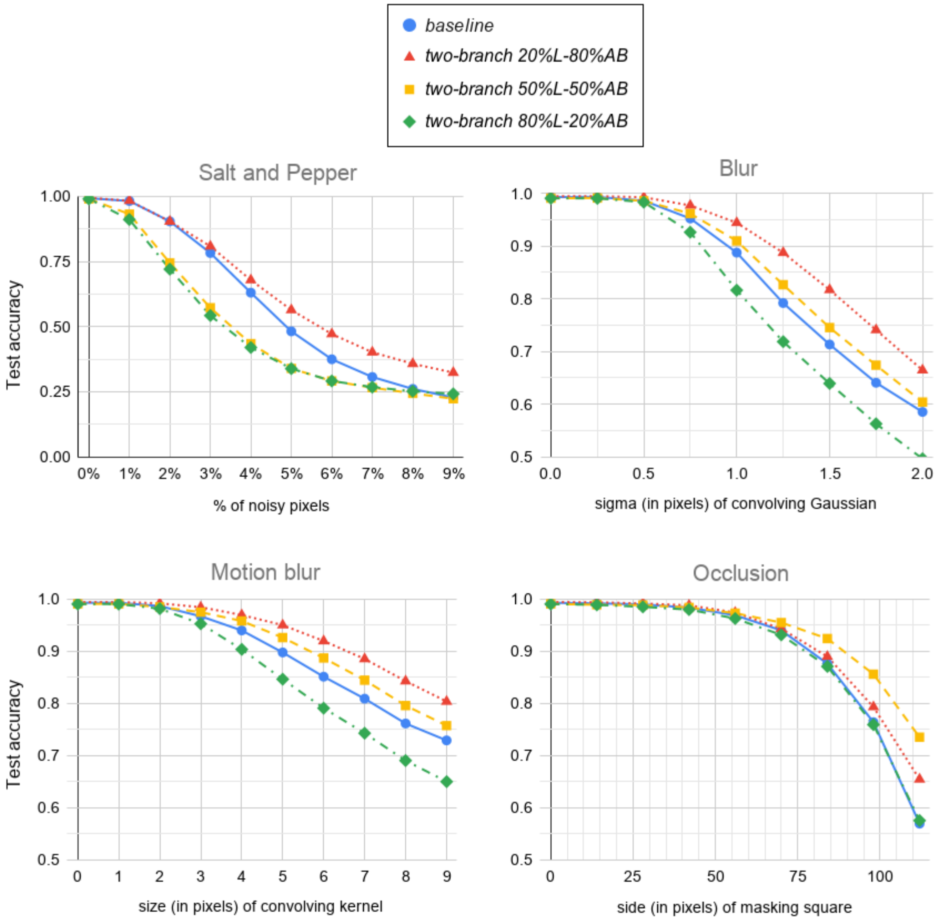

Figure 2. Result plots showing the test accuracy evolution of four approaches under a range of perturbation with four types of noise.

In Blur experiments, a Gaussian distribution of a given sigma in image space coordinates (distance in pixels) is convolved with the input RGB image values producing the typical blurring effect (check Fig. 3). This type of noise simulates unfocused snapshots or dirty lenses. In the corresponding plot, our 20%L-80%AB variant proves the most reliable under the tested range of sigmas. From $\sigma = 1.25$ to $\sigma = 1.75$, this best model overcomes the baseline by 10% of test accuracy. Moreover, the 50%L-50%AB variant also overcomes the baseline, although by a slight difference.

Motion blur is similar to blur (also check Fig. 3), but instead of a Gaussian distribution we use a sparse matrix of a given size with all cells equal to zero except for one line of cells, which is filled with ones divided by the number of cells in that line. By convolving the image pixel values with such a matrix (kernel), it is possible to simulate the blurring due to sudden camera shifts. The direction of movement is parallel to the line of cells different to zero. The extend of movement is equivalent to the length of that line. The corresponding plot depicts similar behavior to the blurring plot, although it is

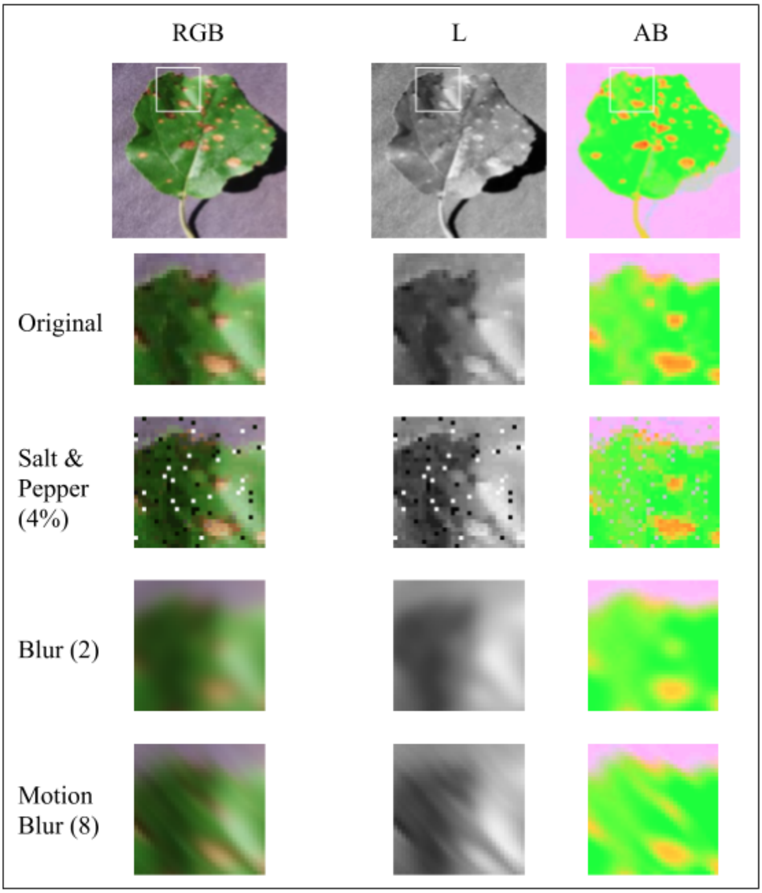

Figure 3. Noise injection in a portion of a test image (Apple Black Rot num.5), in RGB, L and AB spaces: Salt & Pepper noise in 4% of the image pixels; Blur by convolving a Gaussian bell with $\sigma = 2$ pixels; Motion Blur in up-left direction with 8 pixels of kernel width.

necessary to use a 9 pixels-side kernel to degrade the test accuracy of the 20%L-80%AB variant as much as with a $\sigma = 1.5$ in the blurring experiment.

Occlusion is performed by overlapping a square of gray pixels of a given size in a random position of the image. This type of noise simulates the occlusion of the target leaf by other non-interesting objects such as tree branches, fruits, etc. For these experiments, the model that renders the best reliability in the corresponding plot is our 50%L-50%AB variant with a remarkable difference of 5% above the second best model, the 20%L-80%AB variant, which in turn is also 5% above baseline and the 80%L-20%AB variant when the side of the masking square is beyond 100 pixels.

5. Digression

All results are highly determined by the fact that the leaf shape and their lesions are less varying in AB channels than in RGB and L channels as can be seen in the example of

Figure 3. In other words, the leaf representation in AB channels render broad areas of similar colors. This low-frequency nature of the AB channels makes the color-trained filters to inherently take into account a wider field of view. Therefore, more erroneous pixels are needed to mislead the classification. In contrast, the same leaf surface renders more frequent variations in RGB channels which provokes that their trained filters will have a smaller field of view. Specifically, high-frequency noise affects more to gray-level filters, which are actually the ones projected into the L channel. These observations may explain why focusing 80% of the filters on the AB branch provides the best results in presence of most types of noise.

For salt and pepper noise, the effect of spurious pixels in AB channels is noticeable but the larger field of view of corresponding filters allows to overcome those perturbed values. In the other hand, the field of view of L and RGB filters is closer to the area of each erroneous pixel. However, the baseline is more reliable than 50%L-50%AB and 80%L-20%AB configurations because its filters can better treat the spurious changes in the 3D RGB space than the combination of the split L and AB filters.

In contrast to salt and pepper, blurring is a perturbation of low-frequency nature. Despite this fundamental difference, our 20%L-80%AB configuration becomes again the most reliable. In this case, the smoothing of pixel values degrades more the features encoded in the L and RGB channels than the features encoded in the AB channels. The 50%L-50%AB configuration is also stronger than the baseline. In regards to motion blurring, the 20%L-80%AB and 50%L-50%AB configurations are again the most reliable.

For the occlusions experiment, the 50%L-50%AB and 20%L-80%AB variants are the most resilient specially for mask sizes above 1/4 of the total image area. Again, the reasoning for this effect is that a big occlusion in the AB image removes less relevant details than the same occlusion in L and RGB images as the key features in AB channels are wider in image space than in L or RGB channels.

6. Conclusion

In this paper, we have suggested a two-branch CNN for plant disease classification where the first three convolutional layers are specialized in learning chromatic and achromatic features from the CIE Lab color space. Besides classifying original RGB images with similar accuracy and less weights, our experiments also show that our 20%L-80%AB and 50%L-50%AB models better classify input images under salt and pepper, blurring, motion blurring and occlusion by margins up to 10%.

With regards to the optimal distribution of filters among achromatic and chromatic branches, our experiments show that about 80% of the filters should go into the chromatic branch in order to provide the maximum reliability in front of different sources of noise. The reason behind this conclusion is based in the fact that color filters have wider field of view than lightness or RGB filters. Another reason is the color cue portrays highly relevant features for plant disease classification.

As we have Toda & Okura's Inception V3 based work as our baseline, we did our experiments with a modified Inception V3. It would make sense as a future work to try the same two-branches approach with an Inception V4 [13] model.

References

[1] Wang M. Multi-path Convolutional Neural Networks for Complex Image Classification. CoRR. 2015;abs/1506.04701. Available from: http://arxiv.org/abs/1506.04701.

[2] Chen Y, Li J, Xiao H, Jin X, Yan S, Feng J. Dual Path Networks. CoRR. 2017;abs/1707.01629. Available from: http://arxiv.org/abs/1707.01629.

[3] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition; 2016. Available from: http://arxiv.org/abs/1512.00567.

[4] Hughes DP, Salath'e M. An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. CoRR. 2015;abs/1511.08060. Available from: http://arxiv.org/abs/1511.08060.

[5] Schler JPS. Optimizing CNNs first layer with respect to color encoding. In: Valls CJA, editor. 6th URV Doctoral Workshop in Computer Science and Mathematics. vol. 1. Universitat Rovira i Virgil. Tarragona, Catalunya, Spain: Universitat Rovira i Virgil; 2020. p. 4.

[6] Krizhevsky A. Learning multiple layers of features from tiny images; 2009.

[7] Ferentinos KP. Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture. 2018;145:311 318. Available from: http://www.sciencedirect.com/science/article/pii/S0168169917311742.

[8] Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Computational Intelligence and Neuroscience. 2016 Jun;2016:3289801. Available from: https://doi.org/10.1155/2016/3289801.

[9] Mohanty SP, Hughes DP, Salath M. Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science. 2016;7:1419. Available from: https://www.frontiersin.org/article/10.3389/fpls.2016.01419.

[10] G G, J AP. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering. 2019;76:323 338. Available from: http://www.sciencedirect.com/science/article/pii/S0045790619300023.

[11] Toda Y, Okura F. How Convolutional Neural Networks Diagnose Plant Disease. Plant Phenomics. 2019 03;2019.

[12] Pouli T, Reinhard E, Cunningham DW. Image Statistics in Visual Computing. 1st ed. USA: A. K. Peters, Ltd.; 2013.

[13] Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning; 2017. Available from: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806.