




Simplifying VGG-16 for Plant Species Identification

Juan Campos , Arturo Yee , and Ines F. Vega 

Abstract—Plant species identification represents an extraordinary challenge for machine learning due to visual interspecies similarities and large intraspecies variations. Furthermore, research literature reports that plant species identification usually lacks sufficiently large datasets for training classification models. In this paper, we address this problem with a model that simplifies the VGG-16 architecture, the N -VGG model. The idea behind N -VGG is to reduce experimentally observed overfitting on VGG-16 by using as few trainable parameters as possible. To do this, we substitute the flattening layer on the VGG architecture with a global average pooling layer. This reduces the size of the feature vector. In addition, we eliminate one of the two fully-connected layers and use a new hyper-parameter, N , to indicate the number of nodes on the remaining layer. To show the robustness of the N -VGG model, we conducted extensive experimentation. We trained N -VGG on five datasets for plant species identification. Four of these datasets are publicly available and have been widely used as benchmarks for plant identification models. For all datasets, we compare the accuracy of N -VGG to that of the VGG-16, Inception-v4, and EfficientNet-B3 models. The experimental results show that the N -VGG model achieved the best classification performance for all but one datasets, whereas all the models showed a remarkable performance for the remaining dataset. This evidence supports our initial idea that, for plant species classification, some accuracy might be lost due to overfitting and that having fewer trainable parameters helps in producing a more robust model.

Index Terms—Convolutional Neural Network, Deep Learning, Fine-Grained Classification, Plant Species Identification, VGG-16

I. INTRODUCTION

In recent years, deep learning techniques have become quite popular due to their outstanding success in many image classification challenges [1] [2] [3]. Contrary to conventional machine learning techniques, in which features are chosen manually and extracted through carefully crafted algorithms, deep learning techniques automatically discover increasingly higher-level features from data. A convolutional neural network (CNN) is a type of deep learning technique commonly used for image detection, recognition, and classification.

The VGG-16 architecture is one of the most widely used CNN architectures for image classification. Generally, the models based on VGG-16 have obtained good accuracy [4] [5] [6] [7]. Although it is not the newest architecture, it has shown to be a powerful feature extractor. In fine-grained image classification, feature extractors play a key role in the success of a classification model. This is because the model requires

that features capture the most distinguishable traits among different classes while preserving invariant features within the same class. One of the main reasons that fine-grained classification is considered challenging is the difficulty for visually distinguishing classes.

When we build a classification model, the model parameters are modified during the training phase. The goal is to obtain the parameter values that allow the model to perform the classification while minimizing the loss value. Ideally, we want to train the model using a large number of observations per class, but this is not easy in practice. Despite the large availability of data [8], fine-grained classification problems, such as plant species identification, usually deal with datasets that have limited data [9]. When the number of instances in the dataset is small, the model usually overfits. This occurs because models based on deep learning adjust millions of parameters to discover the invariant features in the data. Accessing a large dataset to solve a fine-grained classification problem is not simple. Therefore, techniques such as transfer learning are used so that the model does not estimate parameter values from scratch [4] [10]. This allows the model to use the parameter values acquired in a source dataset as a starting point. Generally, a model that uses transfer learning has a rapid accuracy convergence during the training phase. Another solution is to reduce the number of parameters in the models as much as possible [11].

In this paper, we addressed a multi-class classification problem of plant species with few instances in the dataset. This problem represents an extraordinary challenge for machine learning due to high interspecies similarities and high intraspecies variations in this domain. Our approach is based on modifying the VGG-16 architecture to reduce the number of trainable parameters as well as computing resources without losing accuracy. The proposed modifications are the following. First, the flattening layer in the VGG-16 architecture is replaced by a global average pooling (GAP) layer. Second, the two fully-connected layers of the VGG-16 architecture are replaced by a single fully-connected layer. These modifications reduce the number of trainable parameters from almost 135 million down to approximately 15 million. Our model incorporates an extra parameter (N) in order to increase or decrease the number of neurons in the fully-connected layer.

The rest of this paper is organized as follows. Section II describes the related work that constitutes the state-of-the-art for plant species identification using digital images. Section III presents a detailed description of the proposed model. Section IV describes the used datasets and computing environment. Section V describes the results obtained during the experimental evaluation of our proposal. Finally, in Section VI we present our conclusions and state future research directions.

Juan Campos, Posgrado en Ciencias de la Informacion, Universidad Autonoma de Sinaloa, Culiacan, Mexico. juan.campos@uas.edu.mx

Arturo Yee, Facultad de Informatica Culiacan, Universidad Autonoma de Sinaloa, Culiacan, Mexico. arturo.yee@uas.edu.mx

Ines F. Vega, Parque de Innovacion Tecnologica, Universidad Autonoma de Sinaloa, Culiacan, Mexico. ifvega@uas.edu.mx

II. RELATED WORK

Nowadays, CNNs are one of the most widely used machine learning techniques in computer vision. CNNs have been successfully applied to complex tasks such as object recognition and detection [12] [13] [14] [15]. For instance, the ImageNet [16], the Pascal Visual Object Classes (VOC) [17], the Common Objects in Context (COCO) [18], and the PlantCLEF challenges [19] [20] [21] [22]. The great success of the AlexNet architecture [23] in the ImageNet challenge in 2012 motivated several research groups in computer vision to explore deep learning techniques. AlexNet set the foundations for the development of new CNN architectures such as VGG-16 [24], VGG-19 [24], Inception [25], ResNet [26], Inception-ResNet [27], Xception [28], and MobileNetV2 [29], to name a few.

In 2017, Ghazi [4] used the AlexNet, GoogLeNet, and VGG-16 architectures to generate classification models for plant species identification. The authors trained CNN models both from scratch and using transfer learning. Data augmentation was also applied based on simple image transforms such as extracting and scaling random square regions and image rotations. The authors used transfer learning from a model trained on the LifeCLEF 2015 dataset that contains 113,205 images that correspond to 1,000 plant species native to France and neighboring countries. In the experimental stage, models based on GoogLeNet and VGG-16 achieved better results than the AlexNet model. The authors reported that VGG-16 achieved the best *Top-1* accuracy, with 78.44%. This work obtained second place in the PlantCLEF 2016 challenge.

In 2018, Sulc [30] used Inception-ResNet-v2 and Inception-v4 in the ExpertLifeCLEF 2018 plant identification challenge [22]. The challenge consists of identifying 10,000 plant species from digital images. The training dataset contains 256,288 images validated by experts and 1.4 million noisy images. The authors proposed five different solutions to the challenge. One of the proposed solutions consists of a single Inception-v4 model that obtained an 83.2% *Top-1* accuracy. The rest of the proposed solutions are ensembles of CNN models. With an ensemble of 12 CNN models based on six Inception-ResNet-v2 models and six Inception-v4 models, the authors finished first in the ExpertLifeCLEF 2018 plant identification challenge with 88.4% *Top-1* accuracy.

Lee [31] proposed the combination of a wide and deep learning model for plant species identification. The proposed method combines a linear model and a deep learning model to consider discrete features simultaneously with image content. The linear model used metadata such as flowering dates and geographic coordinates, whereas the deep learning model used the digital image. The authors created a dataset with 14,746 flower images of 100 Korean plants species. They used the Inception-v4 as a baseline model, obtaining 71% in *Top-1* accuracy. On the other hand, the proposed model obtained 78% in *Top-1* accuracy using GPS and date information.

In 2020, Aravin [5] used the AlexNet and VGG-16 architectures for the identification of five diseases affecting eggplants (*Solanum melongena*). The authors used transfer learning and data augmentation to build classification models. Furthermore,

the authors proposed modifications to the AlexNet and VGG-16 architectures. These modifications focused on increasing the number of layers in the fully-connected network to enhance the accuracy of the models. The authors proposed an eggplant diseases dataset of plant leaves images taken in field conditions using smartphones. These images were manually segmented to remove the background. The best model was obtained from VGG-16 which achieved 96.70% of average *Top-1* accuracy.

Saedi and Khosravi [32] in 2020, proposed three customized architectures based on CNN for precision horticulture practices. The authors classified six classes of on-branch fruits. The customized architectures contain two, three, and five convolutional layers, respectively. Furthermore, the authors used the global average pooling layer and the flattening layer in the architectures to compare the number of trainable parameters of the models. Using the global average pooling layer, the authors reported a reduction in the number of trainable parameters in the first fully-connected layer with respect to architectures that used the flattening layer. The authors divided the training into two phases. First, the authors trained model using the ImageNet 2012 dataset. Second, these customized models were used as starting points to re-train using the target dataset. The best results were obtained from the model that uses three convolutional layers. The authors reported 99.76% in *Top-1* accuracy.

Li [6] proposed two classification models for plant diseases classification. The authors used a modified VGG-16 model as a feature extractor. They used the output of one of the first convolutional layers to obtain a 128-dimensional feature vector. The authors called it a shallow CNN extractor. These feature vectors served as the input data for the classification model based on support vector machines (SVM) and random forest (RF). The proposed classification models were trained on the Maize, Apple, and Grape datasets. These datasets are subsets of the Plantvillage [33] dataset. The authors created balanced versions of each dataset. Each dataset has 2,000 images divided into four classes. In their experimental results, the authors reported a Macro F-1 score of 0.94 on all datasets.

In 2021, Vizcarra [34] used the AlexNet, VGG-19, ResNet-101, and DenseNet-201 architectures to classify ten timber-tree species. The authors used transfer learning to train the classification models. They introduced the Peruvian Amazon Forestry dataset that includes 59,441 leaves images from ten of the most profitable and endangered timber-tree species. The image background was removed to eliminate texture noise in the boundary. The results showed that the AlexNet and VGG-19 models outperformed the ResNet-101 and DenseNet-201 models. The VGG-19 model obtained the best accuracy, reaching 96.52% in *Top-1* accuracy.

Bisen [35] proposed an automated plant identification model for identifying plant species using images of their leaves. In this work, the author used a customized architecture to reduce overfitting. The customized architecture contains four convolutional layers and three max-pooling layers. The author used different sizes for filters in the convolutional layers. The layer outputs are flattened into a vector form which is followed by a 128-unit fully-connected layer. The customized architecture ends on a soft-max layer. During the training

phase, data augmentation was used. The author used the Swedish leaf [36] dataset that contains 1,125 leaf images of 15 plant species. For each species, there are 75 images. The model obtained 97% in *Top-1* accuracy.

Dourado-Filho and Calumby [37] performed a comparative study of multiple models based on CNNs to extract deep features from images of multi-organ plant observations. The authors used transfer learning with the InceptionV3, Nas-NetLarge, ResNet50, ResNet152_V2, VGG-16, and VGG-19 architectures. A SVM classifier was used to evaluate the CNNs. The PlantCLEf 2013 dataset [38] was used for experimentation. This dataset contains 26,077 images of 250 plant species from the French flora. The best SVM model yield 0.82 of Micro-F1.

The aforementioned works use either a single model or an ensemble of models to address the plant species identification task. We note that these models need a large number of parameters. We also note that the proposals that modify architectures deal only with a low number of classes. Only a few papers propose simpler models for plant species identification. In contrast, our approach uses a single model based on a simplified VGG-16 architecture to increase accuracy while using as few trainable parameters as possible.

III. DESCRIPTION OF THE PROPOSED CLASSIFICATION MODEL

Recently, the use of CNNs has increased due to the results obtained in computer vision challenges. Furthermore, CNNs have proven their robustness after being tested in different domains. Our proposal is based on the VGG-16 architecture because it has shown good performance for plant species identification.

The VGG-16 architecture contains 13 convolutional layers, two fully-connected layers, and a soft-max layer. This architecture requires almost 135 million trainable parameters. Despite the large number of trainable parameters, the network structure is simple. VGG-16 is divided into a feature extractor and a classifier. The feature extractor is composed of blocks that contain 13 convolutional and five pooling layers. These layers generate an output tensor. This output tensor is processed by a flattening layer to generate a feature vector of size (25088). The feature vector is the input to the second part of the architecture. The classifier is a fully-connected network that contains 120 million trainable parameters. Our modifications focus on reducing the number of trainable parameters of the fully-connected network. A complete description of the model is presented in the following subsections.

A. VGG-16 Architecture

The input to the VGG-16 is a tensor of order three and size $(224, 224, 3)$ representing a color image in RGB format. The output of a convolutional layer is a tensor of order three and size (h, w, d) . The values of height h and width w are the same as the input tensor, whereas the value of d equals the number of applied filters in the convolutional layer. A filter is also a tensor with the same dimensions as the input tensor but different sizes in h and w . In CNN, a convolution is described

as a mathematical operation that receives an input tensor and applies a filter to obtain an activation map of the input tensor. The convolution is performed by sliding a filter over an input tensor, usually starting in the upper left corner, and moving the filter through all positions where the filter is fully adjusted within the limits of the input tensor. The number of filters on the first convolutional layer of VGG-16 is 64. Thus, the size of the output tensor is $(224, 224, 64)$. The max-pooling layers of this architecture use a 2×2 filter and a stride of two. Therefore, the size of the input tensor is reduced by half along the h and w directions. On the last max-pooling layer, the value of both h and w is 7. After each max-pooling layer, the number of filters applied in the convolutional layers is increased by a factor of two. The last convolutional layer in this architecture has 512 filters. In consequence, the size of the final tensor is $(7, 7, 512)$. The flattening layer transforms the final tensor of order three into a tensor of order one. This tensor is called the feature vector, which is the input for the fully-connected network. The two fully-connected layers have 4096 units each. The last layer in the architecture is a soft-max layer with 1000 units because VGG-16 was created to identify 1000 classes.

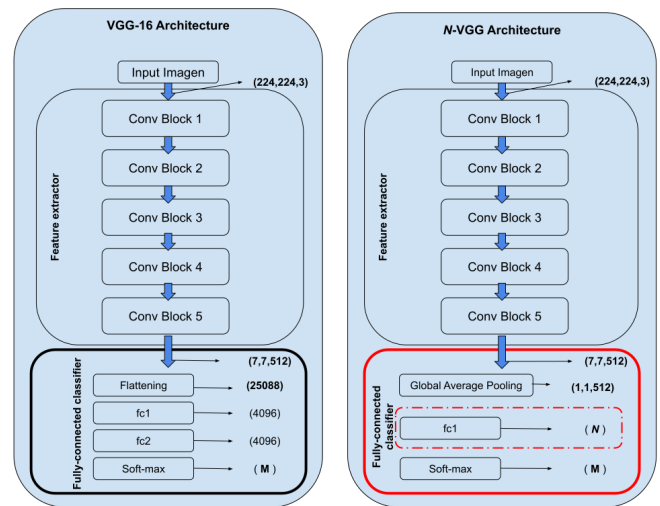


Fig. 1. On the left side is the VGG-16 architecture and on the right side is the N -VGG architecture. N denotes the number of neurons in the fully-connected layer and M is the number of classes.

Our proposal is based on a simplified VGG-16 architecture. This architecture is composed of two parts. We modified the way of generating the input vector of the fully-connected network. We also vary the number of neurons in the fully-connected layer. These modifications significantly reduce the number of trainable parameters. We named our proposal N -VGG.

Figure 1 shows a comparison between the VGG-16 and the N -VGG architectures. In the N -VGG architecture, all the convolutional blocks from the VGG-16 architecture are used. In the classifier, we performed the following modifications.

- 1) A global average pooling layer replaces the flattening layer. The global average pooling layer is based on averaging the output of each activation map in the output tensor. We used the global average pooling layer to

reduce the number of trainable parameters in the model in an effort to reduce overfitting. The flattening layer processes a tensor of size $(7, 7, 512)$ and produces a feature vector of size (25088) . On the other hand, the global average pooling layer transforms the tensor of size $(7, 7, 512)$ into a feature vector of size (512) .

- 2) A single fully-connected layer replaces the two fully-connected layers. We use the parameter N to modify the number of nodes in this layer and reduce the number of trainable parameters. In this way, we have a generic model, that can be trained with different values of the parameter N .

B. Transfer Learning

Transfer learning is the process of reusing the parameter values acquired while training a model in a source dataset and retraining the model in another dataset, called the target. This process uses the parameter values learned while training on the first dataset as the baseline for training a model on a second dataset. Transfer learning can be used in two ways. The trained VGG-16 model can either be used as a feature extractor (i.e., by dropping the fully-connected and soft-max layers), or it can be fine-tuned for improved performance while classifying a second dataset. When the trained model is used as a feature extractor, the parameters of the convolutional layers are frozen. Thus, an input image is processed and a feature vector is produced at the end of the convolutional blocks. The feature vector is then used to train the new classifier. On the other hand, fine-tuning uses the previously trained parameter values of the model as a starting point, and then the model is trained on the new target dataset. During this process, all (complete re-train) or part (partial re-train) of the trainable parameters can be adjusted. In this work, we used a complete re-train when transfer learning was applied.

Transfer learning often brings advantages to classification models. One important advantage is a reduced training time. A second advantage is better performance of the classification models because the model parameter values are not estimated from scratch. Furthermore, transfer learning has proved to have a positive impact on the models' performance when there is not enough data for training [10].

IV. DATASETS DESCRIPTION AND COMPUTING ENVIRONMENT

We compare the results obtained using the N -VGG model to those obtained by the VGG-16 model. We used four publicly available datasets to validate our proposal. In addition, we created our own dataset with images of the Mexican flora.

A. Datasets

We created a dataset with images of the Mexican flora. This dataset started as a compilation of plant images that were later processed to indicate regions of interest. The images were taken from two different sources. In particular, there are images taken by expert biologists during field expeditions in the northwest region of Mexico. These images were complemented with images taken from Naturalista, an online social



Fig. 2. Annotating regions of interest.

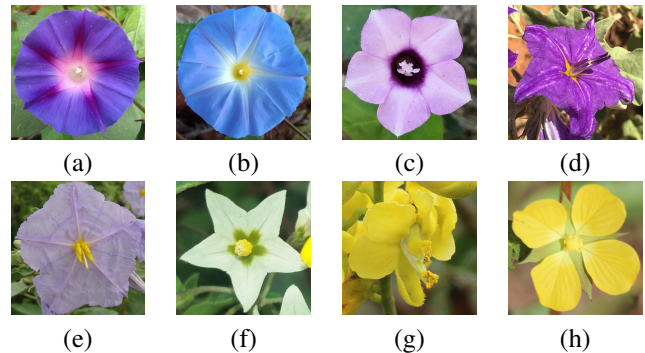


Fig. 3. Images from the Mexico 120 Flower dataset. (a) *Ipomoea purpurea*. (b) *Ipomoea tricolor*. (c) *Ipomoea triloba*. (d) *Solanum tridynamum*. (e) *Solanum hindsianum*. (f) *Solanum americanum*. (g) *Caesalpinia cacalaco*. (h) *Ludwigia octovalvis*.

network for sharing biodiversity information [39]. To focus the attention of the model, regions of interest were defined in each image by manually annotating distinctive organs. For this, we used PitCrop [40], our own image labeling tool. The annotation process was conducted under the supervision of experts in plant taxonomy. Some examples of the regions of interest are shown in Figure 2.

Subsequently, these regions of interest were extracted from the original images to generate a new image for each region of interest. This process required a simple Python script for copying each region of interest from the annotated images into a new image file (this process is known as cropping). The resulting images are always squared. We did this in order to eliminate distortion on the images due to scaling and resizing since the proposed model uses 224×224 RGB images (i.e., larger images are always resized). The dataset consists of 12,000 $m \times m$ color images of plant species from the Mexican flora. These images correspond to 120 plant species. The images contain background, angles, lighting, contrast, and scale variations. Some examples of images in this dataset are shown in Figure 3. In the particular case of this paper, only images of flowers were used. We will refer to this dataset as the Mexico 120 Flower dataset for the rest of this paper.

In addition, we also used the Flavia [41] dataset. This is a well-known dataset containing 1,907 images of leaves corresponding to 32 plant species. All images have white background and have the same size of 1600×1200 pixels. The number of images per species varies between 50 and 77. Figure 4 (a) shows an image from the Flavia dataset. We did not perform any preprocessing on the images in this dataset.

The Swedish leaf [36] dataset contains leaf images of 15 plant species. There are 75 images for each species and a

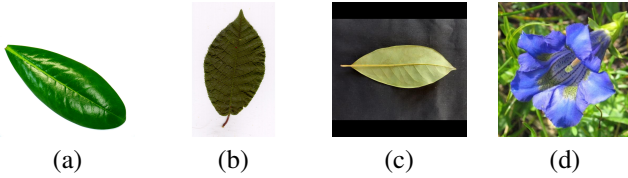


Fig. 4. In our experiments, we used four publicly available datasets. These are examples of the images in (a) Flavia, (b) Swedish leaf, (c) Peruvian Amazon Forestry, (d) Oxford 102 Flower datasets.

total of 1,125 images in the dataset. Figure 4 (b) shows an image from this dataset. These images did require some preprocessing. We transformed the rectangular images into squared images. For this, we took the larger side of the original image and created a white square of the same size. We then superimposed the original image on this white square. The images that resulted from this transformation constitute the dataset used in our experiments.

The Peruvian Amazon Forestry dataset [34] is a collection of 59,441 leaf images of ten timber-tree species collected from the Allpahuayo-Mishana National Reserve, Peru. For each species, there are between 4,344 and 7,494 images. The image examples are dark-background photos taken using six different commercial cameras. Figure 4 (c) shows an image from this dataset. We did not perform any preprocessing on the images in this dataset.

The Oxford 102 Flower [42] dataset contains 8,189 images of flowers of 102 plant species commonly found in the United Kingdom. For each species, there are between 40 and 258 images. These images have large scale, pose, and light variations. Figure 3 (d) shows an image from the Oxford 102 Flower dataset. We noticed that the flowers in the images are centered in most cases. We preprocessed this dataset by cropping, from the original image, the largest inscribed square. The resulting images were used in our experiments.

Table I shows the details of the datasets used for this work. We indicate the distinctive organ used for identification, the number of classes, and the total number of images for each dataset. We have also added a column indicating the number of images per class.

TABLE I
DETAILS OF THE DATASETS USED FOR EXPERIMENTS.

Dataset	Distinctive Organ	Classes	Images	Images per class
Peruvian Amazon	Leaf	10	59,441	4,344 - 7,494
Swedish Leaf	Leaf	15	1,125	75
Flavia	Leaf	32	1,907	50 - 77
Oxford 102 Flower	Flower	102	8,149	40 - 258
Mexico 120 Flower	Flower	120	12,000	100

B. Computing Environment

All the experiments reported in this work were conducted on a PC Workstation with the following specifications. An Intel Xeon W-2133 processor with 32 GB of RAM and an NVIDIA GTX 1080 Graphics Processing Card with 8 GB of memory. The operating system was Linux Ubuntu 18.04. The

software libraries controlling the GPU were provided by the CUDA toolkit 10.0. We used Python 3.6 and Keras 2.2.4 with Tensorflow 1.13.1 as backend to train the deep convolutional neural network architectures.

We used the stochastic gradient descent training algorithm with a learning rate of 1×10^{-4} and momentum of 0.9. We used a batch size of 16 and let the algorithm iterate for 20 epochs. As the loss function we used categorical cross-entropy. We used 10-fold cross-validation to ensure that the models were properly evaluated. The *Top-K* accuracy was used as an index to compare the performance of the models in our experiments. The *Top-K* accuracy refers to the percentage of correct responses in the set of the *K* highest-ranked responses provided by the model.

V. EXPERIMENTAL EVALUATION

We used the VGG-16 architecture as a benchmark to evaluate our proposed model. For the architecture we propose, we experimented with several values of *N*, the number of nodes in the fully-connected layer. This is a hyper-parameter that allows us to fine-tune the model. The following values of *N* were used {8, 16, 32, 64, 128, 256, and 512}. In all cases, we performed a complete fine-tuning during the training phase, starting with the parameter values from a VGG-16 model that was trained on the ImageNet 2012 dataset. We also compare our model to the Inception-v4 and the EfficientNet architectures. Inception-v4 has been successfully used before on similar plant species identification tasks [22] [30] [31], whereas EfficientNet is an architecture proposed to study the effects of scalability on deep convolutional networks [43], which is related to what we do with the hyper-parameter *N*. For both architectures we did a complete fine-tuning starting from models trained on the ImageNet 2012 dataset.

A. Comparison between VGG-16 and N-VGG

The evolution of the *Top-1* accuracy of three models as they are being trained on the Mexico 120 Flower dataset is shown in Figure 5. We observe that this model is overfitting. This is inferred from the plot since the *Top-1* accuracy in the training set (the red curve) reached 100% while the *Top-1* accuracy in the validation set (the blue curve) just reached 83.58%. There is a large gap between the two curves, suggesting overfitting. We acknowledge that this model has too many parameters and hypothesize that this is the source of the overfitting problem. Therefore, by reducing the number of trainable parameters, we should minimize the overfitting and, in consequence, train a model with better *Top-1* accuracy. One way to achieve this on the VGG-16 architecture is by adding a dropout layer after the flattening layer [44]. This randomly eliminates elements of the feature vector, which reduces the number of input nodes on the fully-connected layer and, in consequence, the number of trainable parameters on the model. Figure 5 (b) shows the *Top-1* accuracy of the VGG-16 model using a dropout layer. We should point out that this model took longer to converge than both VGG-16 and *N*-VGG. Therefore, we increased the number of training epochs to 40. In the experiments, we used a dropout rate of 0.5. This reduces the total number of trainable

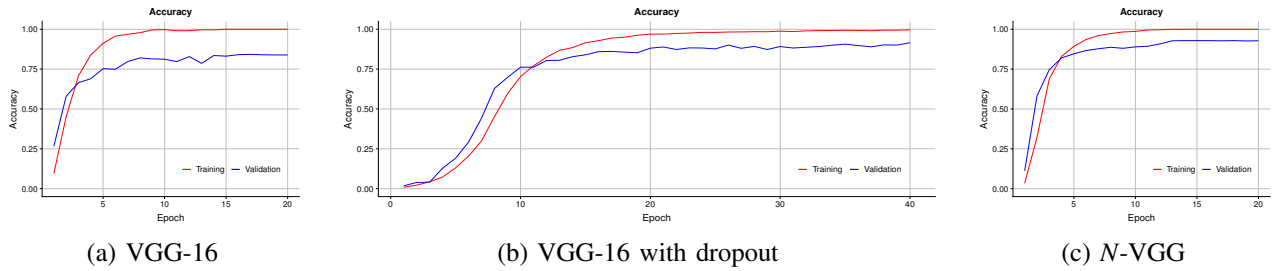


Fig. 5. Comparison of model accuracies during the training phase using the Mexico 120 Flower dataset. (a) VGG-16. (b) VGG-16 with dropout. (c) N -VGG with $N = 512$.

parameters from almost 135 million down to 84 million. We note that the gap between the two $Top-1$ accuracy curves is smaller than the gap between the curves of the VGG-16 model. This model reached 89.32% $Top-1$ accuracy. While these results are considerable better, and provide evidence to support our hypothesis, the number of trainable parameters is still quite large when compared to our proposal.

The observed $Top-1$ accuracy for the N -VGG models for $N = \{256, 128, 64, 32, 16\}$ was similar to the $Top-1$ accuracy for the 512-VGG model shown in Figure 5 (c). We chose not to show their plots as they are redundant. The 512-VGG model showed the smallest gap between training and validation curves. Furthermore, this model had the best $Top-1$ accuracy during the validation of the training phase.

is evidence that having fewer trainable parameters helps in reducing overfitting and produces more robust models. These results also show that too few parameters can have a negative impact on the $Top-1$ accuracy of the model. Such is the case of the performance of 8-VGG.

In Table II, we show the total number of trainable parameters for all the models used in our experiments. We also include each model's $Top-1$ and $Top-5$ accuracies using the Mexico 120 Flower dataset. The first row of Table II summarizes the information of the VGG-16 model. The rest of the rows report the information of the N -VGG model using different values of N . Our proposal reduces the number of trainable parameters from almost 135 million down to 14 - 15 million. We reason that using a single layer in the fully-connected network with a reduced number of nodes (between 8 and 512) produces more accurate models due to less overfitting.

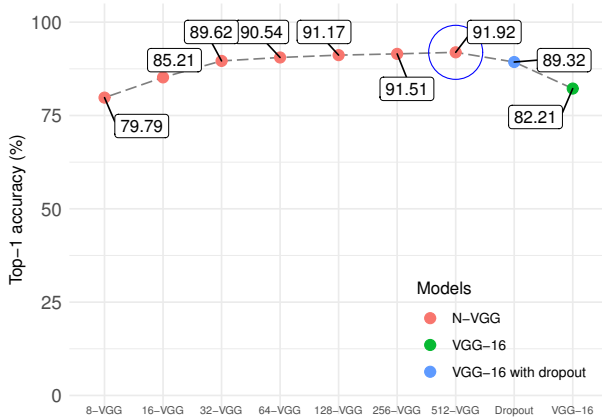


Fig. 6. $Top-1$ accuracy for N -VGG and VGG-16 models using the Mexico 120 Flower dataset.

The performance evaluation of the resulting models is shown in Figure 6. The best performance was obtained by the 512-VGG model, reaching 91.92% for $Top-1$ accuracy. Furthermore, six of the seven N -VGG models we experimented with obtained better $Top-1$ accuracy than the VGG-16 model, which obtained 82.21%. The VGG-16 model with a dropout layer obtained a $Top-1$ accuracy of 89.32%, outperforming the plain VGG-16 model and only the two smallest N -VGG models. Our experiments showed that for N values between 32 and 512, the N -VGG model performs better than both the plain VGG-16 and the VGG-16 with dropout models. As indicated in Table II, we should note that all of the N -VGG models have only approximately 15 million trainable parameters. This

B. N -VGG Models' Performance on other Plant Species Identification Datasets

To evaluate the robustness of our proposal, we also trained models using the Peruvian Amazon Forestry [34], Swedish leaf [36], Flavia [41], and Oxford 102 Flower [42] datasets. We used the same experimental setup described in Section IV-B.

Table III shows the performance evaluation when models were trained using all datasets. In this case, we show the results for VGG-16, N -VGG, and VGG-16 with a dropout layer. For the Peruvian Amazon Forestry dataset, we observed exceptional accuracy for all the models, whereas for the Swedish leaf and Flavia datasets, N -VGG was slightly better than both VGG-16 and VGG-16 with dropout. For both the Oxford 102 Flower and the Mexico 120 Flower datasets, the best performance was obtained by the 512-VGG.

There are almost 135 million of trainable parameters in the VGG-16 architecture, and 84 million when a dropout layer is used. Our proposal, the N -VGG model, requires significantly less, between 14 to 15 million, trainable parameters for $8 \leq N \leq 512$. Our experiments show that this significant reduction in the number of trainable parameters has no negative impact on the $Top-1$ accuracy of the model for all datasets used in our experiments.

C. Comparison to other CNN Architectures

To further validate our proposal, we compared N -VGG to Inception-v4 and EfficientNet. These two models have fewer

TABLE II

COMPARISON OF THE VGG-16 MODEL AND THE N -VGG MODELS, WITH RESPECT TO THE NUMBER OF TRAINABLE PARAMETERS AND ACCURACY.

Model	Convolutional Layers	Fully-connected Layers	Soft-max Layer	Total	Accuracy (%)	
					Top-1	Top-5
VGG-16	14,714,688	119,545,856	491,640	134,752,184	82.21	95.62
512-VGG	14,714,688	262,656	61,560	15,038,904	91.92	98.65
256-VGG	14,714,688	131,328	30,840	14,876,856	91.51	97.55
128-VGG	14,714,688	65,664	15,480	14,795,832	91.17	98.27
64-VGG	14,714,688	32,832	7,800	14,755,320	90.54	97.37
32-VGG	14,714,688	16,416	3,960	14,733,744	89.62	97.14
16-VGG	14,714,688	8,208	2,040	14,724,936	85.21	94.32
8-VGG	14,714,688	4,104	1,080	14,719,872	79.79	91.65

TABLE III

COMPARISON OF *Top-1* ACCURACY OF THE N -VGG, VGG-16, AND VGG-16 WITH DROPOUT MODELS. THE NUMBER OF TRAINABLE PARAMETERS HAS BEEN ROUNDED TO THE NEAREST MILLION.

Model	Number of trainable parameters (millions)	Peruvian Amazon Forestry	Swedish leaf	Flavia	Oxford 102 Flower	Mexico 120 Flower
VGG-16	135	99.56	99.36	98.27	78.38	82.21
VGG-16 with dropout	84	99.76	98.68	98.01	84.86	89.32
512-VGG	15	99.69	99.55	99.14	92.31	91.92
256-VGG	15	99.72	99.47	99.16	91.86	91.51
128-VGG	15	99.70	99.62	99.21	91.17	91.17
64-VGG	15	99.70	99.29	99.20	90.51	90.54
32-VGG	15	99.68	98.62	98.79	89.60	89.62
16-VGG	15	99.68	98.94	88.24	85.99	85.21
8-VGG	15	99.62	85.77	82.42	77.06	79.79

trainable parameters than VGG-16. EfficientNet is a scalable architecture and several scale factors were proposed by its authors. In our case, we use the scale factor known as B3, since it produces an architecture similar in size (with respect to the number of trainable parameters) to our proposal. The resulting architecture is known as EfficientNet-B3. It has 12 million trainable parameters and its input is a color image of size 300×300 . To train the Inception-v4 and EfficientNet-B3 models, we used the same experimental setup described in Section IV-B. During our experiments, we found out that EfficientNet-B3 converges slower than Inception-v4. Therefore, we increased the number of epochs from 20 to 40.

Table IV shows the *Top-1* accuracies of the models evaluated on the Peruvian, Swedish leaf, Flavia, Oxford 102 Flower, and Mexico 120 Flower datasets. In this case, we are only using 128, 256, and 512 as values for N , since they produced the most accurate results. We observed excellent performance by all the models on the leaf-based datasets (e.g., Peruvian, Swedish, and Flavia). These datasets are relatively simple, with a small number of classes and images taken under controlled conditions. In summary, classifying flat leaves is not a challenging problem for CNNs.

From the experimental results, it seems that the real challenge is on classifying the flower-based datasets. These two datasets have a considerably larger number of classes. In addition, images were taken under field conditions, with wide variations on backgrounds, lighting, perspective, and scale. On these two datasets, the 512-VGG model produced the highest accuracy, followed closely by EfficientNet-B3. It is worth noting that VGG-16 with dropout achieved a higher performance than the plain VGG-16 model. Also, while Inception-v4 performed better than VGG-16 on these datasets, it was

surpassed by both EfficientNet-B3 and N -VGG. These results support our idea that a reduction on the number of trainable parameters yields better models as it prevents overfitting.

VI. CONCLUSION

In this work, we proposed a modification to the VGG-16 architecture to address a plant species identification problem. The proposal is based on the simplification of the VGG-16 architecture to enhance the accuracy of the models. We proposed a simplification because the VGG-16 model showed overfitting due its the large number of trainable parameters. We hypothesize that, by reducing the number of parameters we can reduce the overfitting and, in consequence, generate a robust model with better precision. We named our proposal N -VGG. To provide evidence that our proposal is robust, we experimented with five datasets; four publicly available datasets and widely used as benchmarks for plant species identification, and our own Mexico 120 Flower dataset.

In the experimental results, we noticed some peculiarities. The models that were trained using the leaf-based datasets (e.g., Peruvian, Swedish, and Flavia) had accuracy values over 99%. We identify the following three reasons for this behavior. First, the three datasets contain only leaf images that were taken under controlled conditions. Second, the leaf is always in the same perspective on the image and with uniform background. Third, the number of classes in these datasets is small compared to the flower-based datasets (e.g., Oxford 102 and Mexico 120).

For the models that were trained using the flower-based datasets, we observed a lower accuracy than the accuracy observed for the leaf-based datasets. This behavior can be explained as follows. These datasets have a significantly

TABLE IV
COMPARISON OF *Top-1* ACCURACY OF THE *N*-VGG, INCEPTION-V4, AND EFFICIENTNET-B3 MODELS.

Model	Number of trainable parameters (millions)	Peruvian Amazon Forestry	Swedish leaf	Flavia	Oxford 102 Flower	Mexico 120 Flower
Inception-v4	41	93.34	95.09	98.62	87.15	86.96
EfficientNet-B3	12	99.78	98.75	98.42	90.11	91.56
512-VGG	15	99.69	99.55	99.14	92.31	91.92
256-VGG	15	99.72	99.47	99.16	91.86	91.51
128-VGG	15	99.70	99.62	99.21	91.17	91.17



Fig. 7. Images of *Bidens odorata* taken from the Mexico 120 Flower dataset. There are large variations in background, perspective, illumination, and shape of the flowers, making its classification an extraordinary challenge.

larger number of classes than the leaf-based datasets. In addition, the images in these datasets correspond to flowers (a three-dimensional organ), and they were taken under field conditions, which implies large variations of background, perspective, lighting, contrast, and scale. We acknowledge that these conditions negatively affect the feature learning process of the convolutional networks and, consequently, degrade their accuracy. To illustrate how challenging the task of classifying images of flowers taken under field condition can be, we present Figure 7. In this figure we include five images of *Bidens odorata*. While all images correspond to flowers of this plant, the background, focus, perspective, and even the shape of the flowers vary drastically from image to image. Furthermore, the Mexico 120 Flower dataset presents high interspecies similarities and high intraspecies variations, thus making its identification an extraordinary challenge for deep learning models, where fine details make the difference between one species and another, while at the same time, there are significant differences between individuals of the same species.

Our experiments show that the *N*-VGG model can compete with models based on more recent architectures such as the Inception-v4 and EfficientNet-B3 models. Our results also provide evidence suggesting that simple architectures such as *N*-VGG, EfficientNet, or even VGG-16 with dropout can train robust models. Overall, we provide extensive empirical evidence indicating that a simple and efficient model could effectively address a fine-grained classification problem such as the plant species identification task.

For future work we intend to evaluate the accuracy of *N*-VGG when the number of classes increases. Furthermore, we would like to evaluate the effect of transfer learning from different domains on the accuracy of the models.

ACKNOWLEDGMENT

The authors wish to thank the Mexican Council of Science and Technology (CONACYT) for the scholarship awarded to the first author. They would also like to thank for the

financial support provided by the research grant 291772 from the CONACYT-INEGI fund.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning.," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [3] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks.," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.
- [4] M. Mehdipour-Ghazi, B. A. Yanikoglu, and E. Aptoula, "Plant Identification Using Deep Neural Networks Via Optimization of Transfer Learning Parameters.," *Neurocomputing*, vol. 235, pp. 228–235, 2017.
- [5] K. R. Aravind, P. Raja, R. Ashiwin, and K. V. Mukesh, "Disease Classification in Solanum Melongena Using Deep Learning.," *Spanish Journal of Agricultural Research*, vol. 17, no. 3, 2019. <https://doi.org/10.5424/sjar/2019173-14762>.
- [6] Y. Li, J. Nie, and X. Chao, "Do we Really Need Deep CNN for Plant Diseases Identification?," *Computers and Electronics in Agriculture*, vol. 178, 2020. <https://doi.org/10.1016/j.compag.2020.105803>.
- [7] S. H. Lee, H. Goëau, P. Bonnet, and A. Joly, "New Perspectives on Plant Disease Characterization Based on Deep Learning.," *Computers and Electronics in Agriculture*, vol. 170, 2020. <https://doi.org/10.1016/j.compag.2020.105220>.
- [8] J. M. Johnson and T. M. Khoshgoftaar, "Survey on Deep Learning with Class Imbalance.," *Journal of Big Data*, vol. 6, no. 1, pp. 27–27, 2019.
- [9] A.-X. Li, K.-X. Zhang, and L.-W. Wang, "Zero-Shot Fine-Grained Classification by Deep Feature Learning with Semantics.," *International Journal of Automation and Computing*, vol. 16, no. 5, pp. 563–574, 2019.
- [10] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of Transfer Learning for Deep Neural Network Based Plant Classification Models.," *Computers and Electronics in Agriculture*, vol. 158, pp. 20–29, 2019.
- [11] F. Chollet, *Deep learning with Python*. Manning Publications, 2017.
- [12] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization.," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 27-30*, pp. 2921–2929, IEEE Computer Society, 2016.
- [13] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification.," *Ecological Informatics*, vol. 40, pp. 50–56, 2017.
- [14] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of Rice Diseases using Deep Convolutional Neural Networks.," *Neurocomputing*, vol. 267, pp. 378–384, 2017.
- [15] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A Comparative Study of Fine-tuning Deep Learning Models for Plant Disease Identification.," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.
- [16] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-scale Hierarchical Image Database.," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, Florida, USA, June 20-25*, pp. 248–255, IEEE Computer Society, 2009.
- [17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge.," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

- [18] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, September 6-12* (D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), vol. 8693 of *Lecture Notes in Computer Science*, pp. 740–755, Springer, 2014.
- [19] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W. Vellinga, R. Planqué, A. Rauber, S. Palazzo, B. Fisher, and H. Müller, "LifeCLEF 2015: Multimedia Life Species Identification Challenges," in *Proceedings of the International Conference of the CLEF Association, Toulouse, France, September 8-11* (J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. J. F. Jones, E. SanJuan, L. Cappellato, and N. Ferro, eds.), vol. 9283 of *Lecture Notes in Computer Science*, pp. 462–483, Springer, 2015.
- [20] H. Goëau, P. Bonnet, and A. Joly, "Plant Identification in an Open-World (lifeCLEF 2016)," in *Proceedings of the Conference and Labs of the Evaluation Forum, Evora, Portugal, September 5-8* (K. Balog, L. Cappellato, N. Ferro, and C. Macdonald, eds.), vol. 1609 of *CEUR Workshop Proceedings*, pp. 428–439, CEUR-WS, 2016.
- [21] H. Goëau, P. Bonnet, and A. Joly, "Plant Identification Based on Noisy Web Data: the Amazing Performance of Deep Learning (LifeCLEF 2017)," in *Proceedings of the Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14* (L. Cappellato, N. Ferro, L. Goeuriot, and T. Mandl, eds.), vol. 1866 of *CEUR Workshop Proceedings*, CEUR-WS, 2017.
- [22] H. Goëau, P. Bonnet, and A. Joly, "Overview of ExpertLifeCLEF 2018: How far Automated Identification Systems are from the Best Experts?" in *Proceedings of the Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14* (L. Cappellato, N. Ferro, J. Nie, and L. Soulier, eds.), vol. 2125 of *CEUR Workshop Proceedings*, CEUR-WS, 2018.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, May 7-9* (Y. Bengio and Y. LeCun, eds.), 2015.
- [25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, June 7-12*, pp. 1–9, IEEE Computer Society, 2015.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 27-30*, pp. 770–778, IEEE Computer Society, 2016.
- [27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, California, USA, February 4*, Association for the Advancement of Artificial Intelligence, p. 4278–4284, AAAI Press, 2017.
- [28] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, July 21-26*, pp. 1800–1807, IEEE Computer Society, 2017.
- [29] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 18-22*, pp. 4510–4520, Computer Vision Foundation / IEEE Computer Society, 2018.
- [30] M. Sulc, L. Pícek, and J. Matas, "Plant Recognition by Inception Networks with Test-time Class Prior Estimation," in *Proceedings of the Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14* (L. Cappellato, N. Ferro, J. Nie, and L. Soulier, eds.), vol. 2125 of *CEUR Workshop Proceedings*, CEUR-WS, 2018.
- [31] J. W. Lee and Y. C. Yoon, "Fine-Grained Plant Identification using Wide and Deep Learning Model," in *Proceedings of the International Conference on Platform Technology and Service, Jeju, Korea (South), January 28-30*, pp. 1–5, IEEE Computer Society, 2019.
- [32] S. I. Saedi and H. Khosravi, "A Deep Neural Network Approach Towards Real-Time on-branch Fruit Recognition for Precision Horticulture," *Expert Systems With Applications*, vol. 159, 2020. <https://doi.org/10.1016/j.eswa.2020.113594>.
- [33] D. P. Hughes and M. Salathé, "An open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics through Machine Learning and Crowdsourcing," *Computing Research Repository*, vol. abs/1511.08060, 2015.
- [34] G. Vizcarra, D. Bermejo, A. Mauricio, R. Z. Gomez, and E. Danderas, "The Peruvian Amazon Forestry Dataset: A Leaf Image Classification Corpus," *Ecological Informatics*, vol. 62, 2021. <https://doi.org/10.1016/j.ecoinf.2021.101268>.
- [35] D. Bisen, "Deep Convolutional Neural Network Based Plant Species Recognition through Features of Leaf," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 6443–6456, 2021.
- [36] O. Söderkvist, "Computer Vision Classification of Leaves from Swedish Trees," Master's thesis, Linköping University, 2001.
- [37] L. A. D. Filho and R. T. Calumby, "An Experimental Assessment of Deep Convolutional Features for Plant Species Recognition," *Ecological Informatics*, vol. 65, 2021. <https://doi.org/10.1016/j.ecoinf.2021.101411>.
- [38] H. Goëau, A. Joly, P. Bonnet, V. Bakic, D. Barthélémy, N. Boujemaa, and J. Molino, "The Imageclef Plant Identification Task 2013," in *Proceedings of the ACM International Workshop on Multimedia Analysis for Ecological Data, Barcelona, Spain, October 22* (C. Spampinato, V. Mezaris, and J. van Ossenbruggen, eds.), pp. 23–28, ACM, 2013.
- [39] Naturalista, "Comisión Nacional para el Conocimiento y Uso de la Biodiversidad." <http://www.naturalista.mx>, 2021. Accessed: 2021-03-20.
- [40] "https://github.com/zemc77/PitCrop."
- [41] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network," in *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, December 15-18*, pp. 11–16, IEEE Computer Society, 2007.
- [42] M. Nilsson and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," in *Proceedings in the Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, December 16-19*, pp. 722–729, IEEE Computer Society, 2008.
- [43] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML, Long Beach, California, USA, June 9-15* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, PMLR, 2019.
- [44] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.



Juan Augusto Campos-Leal received his B.S degree in Computer Systems from the Universidad Autonoma de Occidente in 2008 and his M.S degree in Computer Sciences in 2018. He is currently a Ph.D. student in Information Science at the Universidad Autonoma de Sinaloa. His current research interests include deep learning and machine learning techniques.



Arturo Yee-Rendon received his B.S. degree in Computer Science from the Universidad Autonoma de Sinaloa. He obtained his M.S. and Ph.D. degrees in Computer Science from CINVESTAV-IPN in 2010 and 2015, respectively. Currently, he is a professor of Computer Science at the Universidad Autonoma de Sinaloa. His current research interests include pattern recognition, deep learning techniques, game theory and optimization algorithms (genetic algorithms).



Ines Fernando Vega-Lopez has been a professor of Computer Science at the Universidad Autonoma de Sinaloa (Culiacan, Mexico) since 2004. He received his Ph.D. degree in Computer Science from the University of Arizona in 2004. His current research interests include high performance database systems, large-scale data analytics, and machine learning.