Deep Learning for Plant Recognition

Olaniyi Bayonle Alao
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
olaniyi-bayonle.alao@stud.hshl.de

***Abstract*—Plant recognition is an essential yet challenging task. Leaf recognition is a vital part of plant recognition, and a good extraction of the unique features in different leaves leads to a robust plant recognition system. Until recently, the combination of computer vision and machine learning techniques have been used for classifying plants. However, these techniques require a lot of manual processes for feature extraction. This paper presents Convolutional neural networks (CNN) architectures for automatic feature extraction and classification in plant recognition. Experimental results showed the effectiveness of CNN architectures, most especially ResNet-152, with an accuracy of 100% for plant recognition tasks.**

***Index Terms*—deep learning, plant recognition, CNN, AlexNet, ResNet**

I. Introduction

Identifying different species and types of plants growing in our natural habitat has long been an important task. However, these tasks have not been easy due to the abundance of plant species we have and the fact that many plants have a lot of similar characteristics. Therefore, experts in the field of agriculture have been trying to identify and document these plants correctly. However, experts also find the task of identifying plant species difficult due to their abundance and characteristics.

With many plants on the verge of going into extinction and biodiversity at risk [21], there has been an increase in the need to reduce the task of experts in plant species conservations to be able to identify plants as easy and less time consuming as possible. From a farmer's point of view, it is crucial to have a system in place that can differentiate between desired plants and unwanted plants - weeds. The automatic detection and controlling of weeds with herbicides using robotics will allow for precise farming, which will, in turn, increase the yields from a planting season [3]. This exact treatment for plants with what they need can reduce farmers' investment, potentially leading to greater profits. In addition, the precise use of herbicides on detected weeds can contribute to a reduction in the environmental pollutions caused by the traditional way of spraying herbicides evenly across farmland.

Recently, researchers have trialled the use of machine learning algorithms to identify plant species through leaf images. The use of this technique is possible since expert botanists often use leaf shape, colour and textures amongst other things for plant leaf classification task [9], [27]. However, using computer vision which is a subset of machine learning, for this task has proven to be challenging [10]. These challenges are because the feature used for leaf identifications and classifications were hand-crafted into these algorithms. Hand-crafting these features meant that the algorithm did not learn to do the classification job independently. Hence it is prone to the same errors expert botanists will make.

Deep learning techniques have been explored and have shown tremendous results to solve these challenges with using computer vision for leaf image classification. The success of deep learning has been attributed to the advancement in the capabilities of graphics processing units (GPU), which can now perform computationally expensive deep learning tasks at incredible speeds [4]. The essential advantage of deep learning over the other traditional machine learning techniques is its ability to automatically extract critical features from the input data using the characteristics of its deep neural networks. Several research studies have shown that convolutional neural networks (CNN) are best suitable for image classification jobs out of the deep learning neural network types.

Zhu et al. [28] proposed two-way attention models for further optimisation in discriminative feature learning and part based attention using deep CNN for automatic plant recognition. The results of their method trained with Xception architecture on four different datasets: Malayakew, ICL, Flowers 102 and CFH plant, produced accuracies of 99.8%, 99.9%, 97.2% and 79.5%, respectively.

Pawara et al. [20] compared models trained using local feature descriptors and bags of visual words with different classifiers to deep CNN models on three plant datasets; AgrilPlant, LeafSnap, and Folio. They trained their CNN models using scratch and fine-tuned GoggleNet and AlexNet architectures. Their other models trained using KNN, SVM, and MLP to classify their hand-crafted extracted features. Their studies found that their deep CNN methods outperformed the hand-craft extracted features.

Anubha et al. [2] conducted a study on using conventional image processing and deep learning techniques for plant recognition tasks. Their conventional image processing techniques include using classifiers like random forest and SVM on manually extracted features. For deep learning techniques, the features automatically extracted using CNN based architectures were categorised using classifiers like logistics regression. The study used Folio, Swedish leaf, and Flavia datasets to train their models. They observed that models trained and classified

using CNN based architectures performed better than ones trained using conventional image processing techniques.

Thanks to the capabilities of CNN, a new dawn has been set in making the task of plant recognition relatively easy and robust for interested individuals. Deep learning models are currently helping millions of agricultural professionals and hobbyists in the correct plant species identification through mobile applications like Leafsnap and Pl@ntNet [14], [25].

This paper compares the accuracies of pre-trained AlexNet and ResNet CNN architectures for plant recognition on the Agriplant and Swedish leaf plant datasets. The rest of the paper is structured as follows: Section 2 begins with a description of the theoretical background of CNN. The section continues with a brief overview of the CNN architectures used to train the plant recognition models, the datasets used, and the hardware and software used to train the models. Section 3 of the paper presents the results of the research. Finally, sections 4 and 5 discussed the research results and concluded the paper with propositions for further research.

II. Materials and Methods

A. Deep Learning

In the past, image classifications workflows consist of processes like image pre-processing, feature extractions and finally classification with machine learning algorithms like support vector machine (SVM), random forest (RF), K-nearest neighbour (KNN), amongst others. The shape, colour, texture and vein were among the features extracted from the leaf images using methods like local binary pattern and Gabor filter [6], [16]. Furthermore, shape and colour extraction techniques like scale space, discrete wavelet transform (DWT) and comparing the colour of images to predefined reference colours are used to extract these features [23]. However, the lack of automation of this process made them impractical for broad adoption. Currently, deep learning methods, particularly CNN, are used to classify plants using leaf images [22].

Deep learning is a model to learning for computers that have some of its bases on the understanding of how the human brain learns complex and specific things [26]. Out of the subset of machine learning, deep learning is the most active field [1].

B. Convolutional Neural Networks

Convolutional neural networks are a type of neural network in DL designed for processing grid-like or multi-array kinds of data [5], [15]. These neural networks typically consist of an input image, convolution, pooling and fully connected layers. The networks architecture of CNN allows them to train on deep layered data structures at a fast pace and robust in the correct classification of images [17]. The achievement of these capabilities are possible through the use of four essential components - the use of many layers, local connections, shared weights, and pooling [15].

1) Convolutional layer: This layer performs mathematical operations on the input image to extract the so-called feature map by using a filter - kernel [5]. The feature map is obtained by summing results gathered from the multiplication of pixel by pixel value and kernel values as the function scans the image from the top left corner down to the bottom right of the picture. This operation results in a smaller size version of the input image. Figure 2 shows a representation of this operation.

A padding operation of the original pixel matrix ensures that all pixels, mainly the corner pixels in the input image, participate in multiple convolution/feature detection operations. This is achieved by adding arbitrary numbers of pixels around the boundaries of the original image. For example, padding of two translates to adding two extra pixels around the borders of the original image matrix. The scanning of the image pixels is called stride, i.e. the number of pixels the filter moves over the input image pixels. For example, a stride value of 3 means that the filter moves by 3 pixels over the image.

2) Pooling layer: The pooling layer is the next layer after the convolution layer. The reduction of features extracted from convolutional operations performed in the preceding layer occurs in the pooling layer. For example, Max pooling operation chooses the maximum value within a rectangular quarter while average pooling takes the average of the values in each area [5]. Achieving non-linearity in the learned networks is made possible by an activation function. An activation function is a function that ensures non-linearity in the networks by determining the activation of neurons in the convolutional and pooling layer [17]. Rectified Linear Unit (ReLU) is CNN's most commonly used activation function.

3) Fully Connected Layer: Recognition and classification of extracted data from previous layers occur in this layer. The neurons directly connect to all activated neurons in the pooling layers in this layer. The outputs in this layer result from the direct connection to all activated neurons in the max pooled layer. Figure 1 shows an overview of the layers in a general CNN architecture.

This paper uses a pre-trained CNN architecture because characteristics learned in a pre-trained CNN model on large datasets can be fine-tuned for a new task [29]. Likewise, there is insufficient diversity in the paper's dataset for training a CNN model from scratch. Hence the reason for resolving to a pre-trained CNN architecture - AlexNet and ResNet.

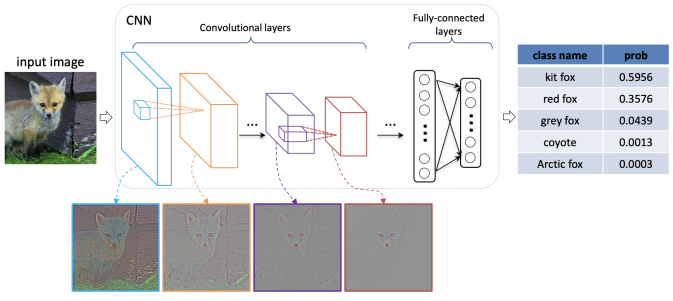

Fig. 1: Screenshot showing an overview of the convolutional and full-connected layers in CNN architecture and the oupt of each layer on the input image [24].

C. CNN Architectures used (AlexNet, ResNet-34)

The paper used the AlexNet and ResNet-34 architectures to train the experiment's model.

AlexNet is a CNN architecture introduced by [13]. AlexNet architecture consists of 8 weight layers: five convolutional layers and three fully connected layers. The first, second and fifth convolutional layers are each followed by normalisation and a max-pooling layer. The first convolutional layer filters the input image with 96 kernels of 11 x 11 in size and a stride of 4 pixels. The second layer filters the input of the first convolutional layer with 256 kernels of size 5 x 5 and a stride of 1 pixel. The third, fourth, and fifth convolutional layers have 384, 256 and 384 kernels with sizes 3 x 3. The fully connected layers have 4,096 neurons each, and the RELU non-linearity function is applied to the output and convolutional layers. The pre-trained network of AlexNet on the ImageNet dataset can classify images into 1000 object categories.

Likewise, ResNet-34 is a CNN network architecture with 34 deeper weight layers introduced by [7]. The ResNet architecture turns a plain neural network (VGG nets but with fewer filters and lower complexity) into a residual network by inserting shortcut connections between the original network layers. The shortcut connections fix the vanishing gradient problem in the deep CNN network by creating an alternative bypass for the gradient by adding outputs from previous layers to that of the stack layers. Figure 2 shows the network architecture of ResNet-34 in comparison with that of plain CNN. Both architectures figure 2 are 34 layers deep, but ResNet has a connection between all the layers. There are different variants of the ResNet architecture. The number in each variant depicts the number of layers in the residual network. Examples are ResNet-18, ResNet-50, ResNet-101, ResNet-110, ResNet-152, ResNet-164, and ResNet-1202.

D. Dataset

The paper performed the experiments using 2 standard datasets; Swedish leaf [24] and AgrilPlant [20].

1) Swedish Leaf Dataset: For plant recognition researches, [24] published the Swedish leaf dataset. This dataset consists of 1125 leaf images from 15 different plant species classes. Thus, each plant species contains precisely 75 images. The tree classes are Ulmus carpinifolia, Acer, Salix aurita, Quercus, Alnus incana, Betula pubescens, Salix alba 'Sericea', Populus tremula, Ulmus glabra, Sorbus aucuparia, Salix sinerea, Populus, Tilia, Sorbus intermedia, Fagus silvatica. The leaf images of each plant species were in a laboratory on a white background. Figure 3 shows sample images from the Swedish leaf dataset.

2) AgrilPlant Dataset: [20] introduced the AgrilPlant dataset for plant recognition tasks. This dataset consists of precisely 300 images of 10 classes of plants, amounting to a total of 3000 images. Flickr website is the source of the images in this dataset. The images may contain the landscape of the entire plant, branch, leaf, fruit and flower. The Agrilplant dataset is very challenging because of the following reasons:

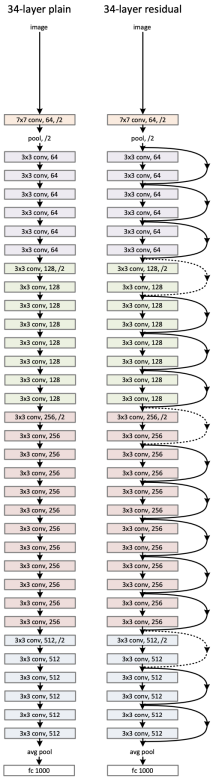

Fig. 2: Screenshot showing an overview of ResNet-34 architecture in comparison with plain CNN architecture [7].

- There are similarities among some classes, i.e. apple, orange, and persimmon, which have similar shapes and colours.
- There is a diversity of plants within the same class. For example, there are green and red apples or varieties of tulips.
- The images contain various objects in the background not relevant to the recognition task. These background noises are from the outdoor environments where the images were taken.
- There is a similarity among some classes. For example, apple, orange, and persimmon images have similar shapes and colours.

  Figure 4 shows sample images from the AgrilPlant dataset.

E. Hardware, Software and Parameter settings

The paper used Jupyter Notebooks [12] running Python 3 kernels with the PyTorch [18] and fast.ai [8] libraries for training and evaluating the CNN models on an Ubuntu 20.04.3 LTS Linux distribution server with 2x AMD EPYC 7452 32-Core Processor, 1 TB of RAM, 7x Nvidia A100-PCI-E-40GB and 1x Nvidia Quadro RTX 5000 graphic cards.

In the experiment, all images in the dataset were rescaled to 256 x 256 pixels for better classification and part localisation. The paper used Adam [11] for optimising the models, learning rate of 0.001, 80% of the dataset for training and 20% for

Fig. 3: Screenshot showing sample images from the classes in the Swedish leaf dataset [24].

testing. Each pre-trained CNN architecture was trained for five epochs on the dataset.

III. Results

1) Swedish Leaf Dataset Evaluation: Table I shows the classification accuracy, training time, and total parameters in the trained CNN architectures on the Swedish leaf dataset. ResNet-50 and ResNet-152 achieved the best performance with an accuracy of 100% in classifying the leaves in the Swedish leaf dataset. However, ResNet-50 had less training time and total parameters than ResNet-152 despite having the same accuracy. On the other hand, AlexNet performed the least in the classification task with an accuracy of 98.7% using the least total parameters compared to other architectures. The AlexNet accuracy reported by Pawara et al. [19] outperforms the AlexNet accuracy in this study. The accuracy of the ResNet-50 and ResNet-152 models in this study exceeds that reported by Anubha et al. [2] using VGG-16 (98.52%) and VGG-19 (99.41%) architectures.

TABLE I: Comaparison of the classification accuracy, training time, and total parameters between CNN methods trained on the Swedish leaf dataset.

Methods	Accuracy	Total Parameters	Training Time (s)
ResNet-18	99.6%	11,711,552	05
ResNet-34	99.1%	21,819,712	05
ResNet-50	100%	25,622,080	08
ResNet-101	99.6%	44,614,208	10
ResNet-152	100%	60,257,856	14
AlexNet	98.7%	2,741,568	06

2) AgrilPlant Dataset: For the AgrilPlant dataset, ResNet-152 obtained the best performance with a classification accuracy of 99.3% and the highest average training time of 28 seconds per epoch. The ResNet-152 model accuracy outperforms the accuracy reported by Pawara et al. [20]. Table II shows the result of ResNet and AlexNet architectures of the datasets.

TABLE II: Comaparison of the classification accuracy, training time, and total parameters between CNN methods trained on the AgrilPlant dataset.

Methods	Accuracy	Total Parameters	Training Time (s)
ResNet-18	98.2%	11,708,992	07
ResNet-34	99.2%	21,817,152	07
ResNet-50	99.2%	25,619,520	14
ResNet-101	99%	44,611,648	20
ResNet-152	99.3%	60,255,296	28
AlexNet	96.3%	2,739,008	04

IV. Discussion

The results of training the plant recognition task models showed ResNet-50 and ResNet-152 to be the best for this task. The high accuracies achieved by the CNN architectures as highlighted in table I is attributed to the fact that the images were taken under controlled laboratory conditions. Despite the AlexNet having the lowest accuracy, its result is remarkable due to having the lowest parameters for training the model. There was some misclassification in prediction when the AgrilPlant and Swedish leaf Plant models were evaluated using random pictures on the internet not present in the training dataset. These misclassifications are due to not enough diversity in the dataset.

V. Conclusion

This paper presents using CNN architectures to detect and classify plant images into individual species. The study used the publicly available Swedish leaf and AgrilPlant datasets for model training. The experiment resized the images to 256 x 256 pixels prior to training with Alexnet and variants of ResNet architectures available in the fast.ai python library. The experiments showed ResNet-152 with an overall accuracy of 100% as the best architecture for plant recognition task on the dataset. In conclusion, CNN architectures showed their effectiveness for use in plant recognition. In future, it will be interesting to research the effect of image transformation techniques like random clipping, rotation and brightness adjustment on the dataset and the overall classification accuracies of trained models.

References

[1] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular systems biology*, 12(7):878, 2016.

[2] S Anubha Pearline, V Sathiesh Kumar, and S Harini. A study on plant recognition using conventional image processing and deep learning approaches. *Journal of Intelligent & Fuzzy Systems*, 36(3):1997–2004, 2019.

[3] M Dian Bah, Adel Hafiane, and Raphael Canals. Deep learning with unsupervised data labeling for weed detection in line crops in uav images. *Remote sensing*, 10(11):1690, 2018.

[4] Sawyer D Campbell, Ronald P Jenkins, Philip J O'Connor, and Douglas Werner. The explosion of artificial intelligence in antennas and propagation: How deep learning is advancing our state of the art. *IEEE Antennas and Propagation Magazine*, 63(3):16–27, 2020.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Fig. 4: Screenshot showing sample images from the classes in the AgrilPlant dataset [20].

[6] Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE transactions on image processing*, 19(6):1657–1663, 2010.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Jeremy Howard et al. fastai. https://github.com/fastai/fastai, 2018.

[9] Abdul Kadir, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa. Leaf classification using shape, color, and texture features. *arXiv preprint arXiv:1401.4447*, 2013.

[10] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. *Jupyter Notebooks-a publishing format for reproducible computational workflows.*, volume 2016. 2016.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[14] Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares. Leafsnap: A computer vision system for automatic plant species identification. In *European conference on computer vision*, pages 502–516. Springer, 2012.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[16] Weitao Li, Kezhi Mao, Hong Zhang, and Tianyou Chai. Selection of gabor filters for improved texture feature extraction. In *2010 IEEE International Conference on Image Processing*, pages 361–364. IEEE, 2010.

[17] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, 2015.

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[19] Pornntiwa Pawara, Emmanuel Okafor, Lambert Schomaker, and Marco Wiering. Data augmentation for plant classification. In *International conference on advanced concepts for intelligent vision systems*, pages 615–626. Springer, 2017.

[20] Pornntiwa Pawara, Emmanuel Okafor, Olarik Surinta, Lambert Schomaker, and Marco Wiering. Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition. In *ICPRAM*, pages 479–486, 2017.

[21] Stuart L Pimm and Lucas N Joppa. How many plant species are there, where are they, and at what rate are they going extinct? *Annals of the Missouri Botanical Garden*, 100(3):170–176, 2015.

[22] Melike Sardogan, Adem Tuncer, and Yunus Ozen. Plant leaf disease detection and classification based on cnn with lvq algorithm. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 382–385. IEEE, 2018.

[23] Manali R Satpute and Sumati M Jagdale. Color, size, volume, shape and texture feature extraction techniques for fruits: a review. *Int. Res. J. Eng. Technol*, 3:703–708, 2016.

[24] Oskar Söderkvist. Computer vision classification of leaves from swedish trees, 2001.

[25] Planet Team. Planet application program interface: In space for life on earth, 2017–.

[26] Haohan Wang and Bhiksha Raj. On the origin of deep learning. *arXiv preprint arXiv:1702.07800*, 2017.

[27] Xue-Yang Xiao, Rongxiang Hu, Shan-Wen Zhang, and Xiao-Feng Wang. Hog-based approach for leaf classification. In *International Conference on Intelligent Computing*, pages 149–155. Springer, 2010.

[28] Youxiang Zhu, Weiming Sun, Xiangying Cao, Chunyan Wang, Dongyang Wu, Yin Yang, and Ning Ye. Ta-cnn: Two-way attention models in deep convolutional neural network for plant recognition. *Neurocomputing*, 365:191–200, 2019.

[29] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.