Conference Paper Title*

[*]Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*— [17] This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

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. But 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 the biodiversity at risk [17], 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 [2]. This exact treatment for plants with what they need can reduce farmers' investment, potentially leading to greater profits. Precise use of herbicides on detected weeds can also 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 [7], [21]. However, using computer vision which is a subset of machine learning, for this task has proven to be challenging [8]. These challenges are due to the fact that the feature used for leaf identifications and classifications were handcrafted into these algorithms. Handcrafting these features meant that the algorithm did not learn how to do the classification job independently. Hence it is prone to the same errors expert botanists will make.

As a solution to these challenges with using computer vision for leaf image classification, the use of deep learning techniques have been explored and have shown tremendous results. The success of deep learning has been attributed to the advancement in the capabilities of graphics processing units (GPU), which can now perform the computationally expensive tasks of deep learning at incredible speeds. 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.

On a positive note, 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 [**?**], [11].

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.

The rest of the paper is structured as follows.

II. Literature Review

A. Maintaining the Integrity of the Specifications

III. 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. Texture features were extracted using methods like local binary pattern, and Gabor filter [4], [13]. 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 [18]. However, the fact that there was no automation of this process made them impractical for broad adoption.

Currently, deep learning methods, particularly CNN, are used to classify plants using leaf images.

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 straightforward things [20]. 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 that is designed for processing grid-like or multi-array kinds of data [3], [12]. These kinds of 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 [14]. 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 [12].

1) Convolutional layer: This layer performs mathematical operations on the input image to extract the so-called feature map by using a filter - kernel [3]. 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 is done to ensure that all pixels, mainly the corner pixels in the input image, participate in multiple convolution/feature detection operations. This is typically 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. Max pooling operation chooses the maximum value within a rectangular quarter while average pooling takes the average of the values in each area [3]. 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 [14]. Rectified Linear Unit (ReLU) is the most commonly used activation function in CNN.

3) Fully Connected Layer: Recognition and classification of extracted data from previous layers occur in this layer. In this layer, the neurons direct connections to all activated neurons in the pooling layers. The outputs in this layer result from the direct connection to all activated neurons in the max pooled layer. Figure 2 shows an overview of the layers in a general CNN architecture.

This paper uses a pre-trained CNN architecture because it is a fact that characteristics learned in a pre-trained CNN model on large datasets can be fine-tuned for a totally new task []. Likewise, there is not enough diversity in the dataset for training a CNN model from scratch. Hence the reason for resolving to a pre-trained CNN architecture - AlexNet and ResNet.

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

The paper used the Alexnet and ResNet-34 architectures for training the model in the experiment.

Alexnet is a CNN architecture introduced by [10]. Alexnet architecture consists of 8 weight layers: 5 convolutional layers and 3 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 layers and that of convolutional layers. The pre-trained network of Alexnet on the ImageNet dataset can be used to classify images into 1000 object categories.

Likewise, the Resnet-34 architecture is a CNN network with 34 deeper weight layers introduced by [5]. 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 x shows the network architecture of Restnet in comparison with that of plain CNN.

D. Dataset

The paper performed the experiments using 2 standard datasets; Swedish leaf [19] and AgrilPlant [16].

1) Swedish Leaf Dataset: For plant recognition researches, [19] 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 x shows sample images from the Swedish leaf dataset.

2) AgrilPlant Dataset: [16] introduced the AgrilPlant dataset for plant recognition tasks. This dataset consists of precisely 300 images of 10 classes of plant, 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:

- 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 there are varieties of tulips.
- The images contain varieties of 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 consist of similar shapes and colours.

  Figure x shows sample images from the AgrilPlant dataset.

E. Hardware and software

The paper used Jupyter Notebooks [9] running Python 3 kernels with the PyTorch [15] and fast.ai [6] libraries for training and evaluating the CNN models on a Linux server with 32 GB of RAM and a GTX 1080 GPU with 8 GB of RAM.

IV. Results

will present the result without explaining why
will compare the result with other research papers

V. Discussion

Will explain the results
the shortcomings
advise for future research

VI. Conclusion

$$a + b = \gamma \tag{1}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

A. Some Common Mistakes

B. Authors and Affiliations

C. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. **??**”, even at the beginning of a sentence.

TABLE I
Table Type Styles

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

References

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

[2] 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.

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

[4] 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.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] 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.