Plant Species Image Recognition using Artificial Intelligence on Jetson Nano Computational Platform

Shruti Chavan, John Ford, Xinrui Yu and Jafar Saniie
Embedded Computing and Signal Processing Research Laboratory (http://ecasp.ece.iit.edu/)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, IL, U.S.A.

Abstract— **The ongoing research for plant/animal species identification by computer vision engineers is exciting and vast. This paper describes a deep learning approach to identify plant species using image analysis. A portion of the LeafSnap dataset, containing 15 plant species and 30 images per species, was used to compare the performance of architectures. Convolutional Neural Networks are trained to capture the features from images and recognize the plant species. An efficient Artificial Intelligence System is designed and implemented with minimal components, including a camera and Jetson Nano (single-board embedded computing device). The experiment used, in particular, CNN architectures- AlexNet, ResNet50, and MobileNetv2, within Python's Tensorflow framework, to accomplish species identification. Of these, AlexNet provided the best results, with 72% validation accuracy after 15 epochs.**

Keywords—convolutional neural network, plant identification, Jetson Nano, LeafSnap dataset

I. INTRODUCTION

Plant researchers, botanists, and scientists identify a particular plant species by looking at the texture, shape, color of the leaves, flowers, fruits, or other parts of the plant. However, it is difficult for those who are unfamiliar with all the plant species to perform this identification. For instance, trekkers and botany students might find a plant species in rarely-visited areas but be unable to identify the rarity of the plant. This causes many potentially-rare plants to remain undiscovered in the wild, impeding plant conservation efforts. With an adequate electronic method of doing so, they can both identify the plant within seconds and help to update and maintain the international standard plants' taxonomic database. Thus, a great demand for AI tools to identify a plant species exists in the present day.

Traditional plant identification approaches are both expensive and time-consuming since they require manual intervention by human experts [1]. Machine learning algorithms are useful but still take longer amounts of time and yield poorer results as compared to deep learning methods. A variety of datasets are available for plant and animal species. These datasets include field ("in the wild") images of wildlife, images of lab-grown wildlife in front of simple backgrounds, or images of naturally-grown wildlife but taken on a simple background. Concentrating our efforts on plants, we looked into the Swedish Leaf Dataset [2], Flavia Leaf Dataset [3], and LeafSnap Dataset [4]. The LeafSnap dataset was used to carry out the experiments. It consists of 23,147 lab images of 185 plants, and the images were obtained by flattening the leaves and putting them on a fixed white background. Training data collected with a simple background often provides better results. Orientation and illumination restrictions were not put on the lab images dataset.

New techniques are available, including programming frameworks such as the MATLAB Image Processing Toolbox, Deep Learning Toolbox, OpenCV, and TensorFlow, which make the species identification task easier and require less human intervention. Users simply capture a photograph of an animal or plant and feed it as input to the system, and the entire description of the corresponding species can be obtained with a single click. Edge computing devices available in the market are used for testing the performance of the design without the Internet. Thus, this experiment aimed at developing a simple CNN architecture that could be easily and efficiently tested on embedded systems. The experiments were carried out using the NVIDIA Jetson Nano, an embedded platform targeted towards low-power AI systems [5]. The Jetson Nano contains a quad-core ARM A57 processor operating at 1.43 GHz and a much more powerful 128-core Maxwell GPU that speeds up the CNN deployment process. The HDMI and USB ports available on the Jetson Nano are utilized for interfacing the camera and display devices.

II. RELATED WORKS

A. *Transfer Learning and Fine-Tuning Approaches*

The Python deep learning API Keras offers easy development for the popular transfer learning approach, in which a pre-trained model is used to draw conclusions on application-specific data. The model could be completely or partially pre-trained using popular image datasets such as ImageNet. This approach is used when the provided dataset is significantly smaller and the CNN deeper. A similar approach is used by authors to make a custom dataset for a project that consisted of six classes of medicinal plants located in Southern India [6]. A user-friendly Android application was designed to capture a video of a plant for twenty seconds and identify it. It used a pre-trained MobileNet model, which was then converted into a compatible TensorFlow Lite model using the TensorFlow Lite converter. The TensorFlow Lite interpreter was used for deploying this module in an Android application. A validation accuracy of 95% was achieved by the authors.

CNN fine-tuning refers to utilizing pre-existing CNN architectures and model weights, making changes only to the last layer in the network. Sungbin Choi performed fine-tuning using GoogleNet on the PlantCLEF dataset [7]. The validation split of 0.2 was set by dividing the dataset into five folds. As a multi-image observation-query was used, different methods were used for score ranking. The author experimented with the Bordafuse method and the majority voting-based method for combining the results. Maximum accuracy was achieved by training 5 CNNs and using a majority-based voting method for score ranking.

B. *Attention Cropping and CNN Training*

Attention cropping refers to isolating a Region of Interest (ROI) from the entire picture. The unwanted background is eliminated and only the region of interest of the image is fed to the neural network. Xiao Qingguo, Guangyao Li, Li Xie, and Qiaochuan Chen proposed a new technique for identifying the plants in a real-time scenario [8]. The design was a combination of attention cropping and convolutional neural networks. Background removal was performed by obtaining image saliency maps. The region of interest (ROI) was cropped with the help of K-means segmentation, and a new dataset was formed in which the non-salient parts of the images were removed. MATLAB was used for image preprocessing and Pytorch was used for training the Convolutional Neural Networks. The experiments involved training the models Inceptionv3 and ResNet50 with and without the aforementioned image augmentation technique. The best result on the Oxford flower dataset, 95%, was achieved using a combination of attention cropping and training with the Inceptionv3 model.

III. PROPOSED METHOD

Many researchers have tried to use Convolutional Neural Networks for feature extraction from gesture datasets. After investigating the existing related designs and studying the available alternatives given several constraints, we can implement a design that is based on a combination of image pre-processing and deep learning to achieve our sought-after objectives. It is easier for users to simply capture a photo of a hand sign and feed it to the system. The task of feature extraction is performed by the CNNs only. Real-time prediction accuracies can be improved by trying transfer learning, CNN fine-tuning methods.

A. *System Description*

The proposed method of plant species identification in this paper is as shown in Fig. 1. The description of certain blocks is also given further.

- *Plant Dataset*: The plant database is a collection of images of various plant species, with multiple images representing a single plant leaf and multiple plant types being incorporated into a single database. The images can be taken in a lab or on a field, depending on whether the network is to be trained to recognize the plants' samples or the plants as seen in their natural environment.
- *Image Processing*: The resolution of images may vary, though when fed to the neural network, they will be standardized to one image resolution via scaling. Training raw unaltered images may result in poor performance. Thus, simple image preprocessing algorithms can be implemented to eliminate noise and ultimately achieve maximum accuracy. Other image processing algorithms, such as RGB to gray conversion, reduce the number of channels and thus reduce the computation and training time and power consumption.
- *Image Augmentation*: Data augmentation assists in working with small databases by artificially increasing database size. These operations result in a greater number of possible variations to the images, allowing modified copies of the images to be created in an effort to 'extend the database' and increase model accuracy.
- *Convolutional Neural Network and Hyperparameters*: Training options for the model are set accordingly before training the database using any CNN architecture. The training options are Maximum batch size, number of epochs, and learning rate.
- *Image Acquisition*: Acquisition can be performed via any camera device, including webcams embedded in desktops, laptops, and smartphone devices. Since all image sizes are standardized and, much more often than not, result in shrinkages with lower resolution images being fed to the neural network, the camera providing the image need not be high-resolution. The recognized name of the species can be displayed in text format along with its common name, morphological area, and distinguishing characteristics.

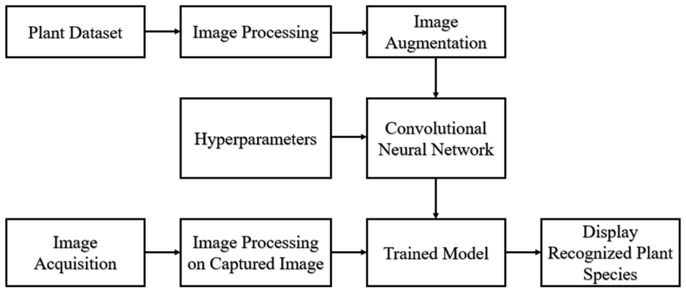
Plant Dataset
Image Processing
Image Augmentation
Hyperparameters
Convolutional Neural Network
Image Acquisition
Image Processing on Captured Image
Trained Model
Display Recognized Plant Species

Figure 1. Block Diagram of Proposed Method

Deep learning is a subsection of artificial intelligence based on creating neural network data representations as opposed to using task-specific algorithms. Learning can be supervised, semi-supervised, or unsupervised. Deep learning architectures, such as deep neural networks, deep belief networks, and recurrent neural networks, have been applied to fields such as computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection, and board game programs, where they have produced results comparable to, and in some cases superior to, human experts. For this reason, deep learning algorithms were employed to complete the task at hand.

In deep learning, an algorithm is fed a set of training images with a known classification and uses those images to adjust each of the weights of the layers in its "guessing" network, depending on the accuracy of the result. A subsection of the training images, known as validation images, is often sectioned off to provide periodic checks with known desired results that do not affect the weights of the network. This is done to ensure that the training is working effectively and is not prone to problems such as being "overtrained" or fed too many images, which would be indicated by a greater accuracy in the testing phase than in the aforementioned validation phase. Test images are images of a classification unknown to the algorithm that the algorithm attempts to classify accurately. If the classification provided by the algorithm matches the one corresponding to the image, the algorithm is said to have accurately classified the image, and the degree to which the algorithm performs correct classifications is known as the algorithm's over-classification accuracy.

B. Hardware Components

As the input to the system is in the form of an image, a camera is required to capture the photograph of the plant. There are no specific requirements regarding the camera, though in the case of the images provided in the LeafSnap database used for this scenario, pictures of plant leaves on a white background were taken with standard mobile devices, such as the iPhone. However, any resolution camera can be used to acquire photos of the subjects to be analyzed.

For the pre-processing of large databases, a fast processor is required. The NVIDIA Jetson Nano embedded system, optimized for machine-learning tasks and portability, was used for both the training and testing in this scenario. In a real-world scenario, with significantly larger databases, training would be performed on a more computationally-capable server, and the weights of each layer derived from the model would then be sent to a device on the edge-side, such as the Jetson Nano, for testing purposes.

C. Software Libraries

Software libraries used include OpenCV for decoding and processing the images and TensorFlow Keras to train the model on the provided dataset.

IV. Implementation

A. Database Preparation

Due to timing constraints during the testing portion, a smaller dataset of 15 randomly-chosen species of plants with 30 images per species was taken out from LeafSnap dataset and became the final dataset used for carrying out the trials. Plant species included in this dataset were as follows: Abies Concolor, Acer Campestre, Acer Ginnala, Acer Griseum, Acer Negundo, Acer Pensylvanicum, Acer Rubrum, Acer Saccharinum, Aesculus Glabra, Aesculus Pavia, Tsuga Canadensis, Amelanchier Arborea, Quercus Virginiana, and Zelkova Serrata. For sample images from this dataset is shown in Fig. 2 [4].

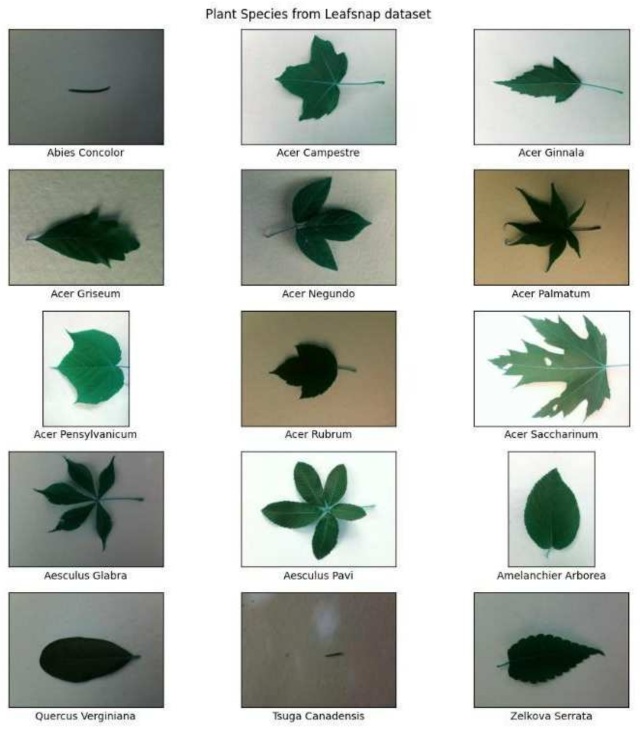

Figure 2. Sample Images from the Subset of LeafSnap Image Dataset

One of the pre-processing algorithms used to increase the recognition accuracy is thresholding. It is a subsection of image segmentation in which a cut-off is set on the overall brightness of each pixel, either rendering it black or white depending on whether or not it meets the cut-off. This results in a two-tone image such as that shown in Fig. 3.

Another pre-processing algorithm intended to increase accuracy is known as image salience mapping, which works by determining the motion performed by the object. The results of applying a saliency map are shown in Fig. 4.

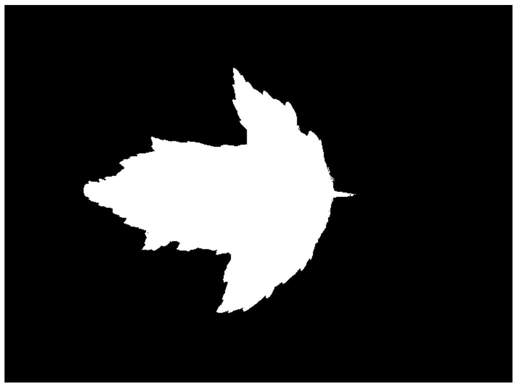

Figure 3. Thresholding Applied on Image of Acer Ginnala

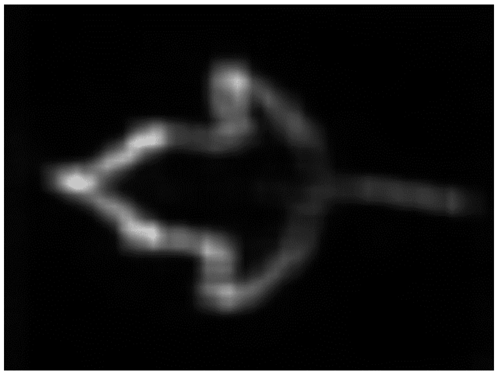

Figure 4. Saliency Detection for Image of Acer Ginnala

V. RESULTS

In this study, the modified database of 15 classes was trained using a simple CNN architecture [9] with some degree of variation in training options (see Fig. 5), to ascertain the ones leading to the maximum possible outcome. The results for training on the simple CNN architecture are listed in Table I. The same database of plant images was then used to train the same simple CNN network but with saliency-detection image preprocessing applied and the results are listed in Table II.

TABLE I. RESULTS FOR TRAINING ORIGINAL DATASET WITH SIMPLE CNN ARCHITECTURE [9]

Parameters	**Number of epochs**	**Training Accuracy**	**Validation Accuracy**
Validation split = 0.2, RGB image dataset	5	80.00 %	65.56 %
	10	92.22 %	63.33 %
	15	99.44 %	71.11 %
Validation split = 0.3, RGB image dataset	5	58.73 %	53.33 %
	10	96.83 %	70.37 %
	15	99.05 %	63.70 %

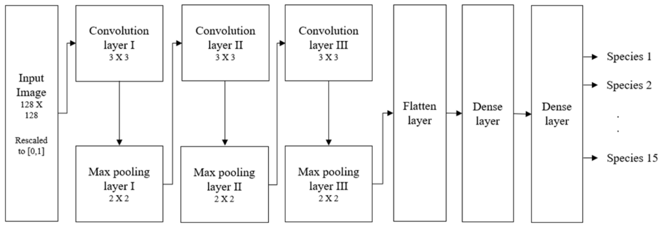
Input Image 128 X 128 Rescaled to [0,1]
Convolution layer I 3 X 3
Max pooling layer I 2 X 2
Convolution layer II 3 X 3
Max pooling layer II 2 X 2
Convolution layer III 3 X 3
Max pooling layer III 2 X 2
Flatten layer
Dense layer
Dense layer
Species 1
Species 2
Species 15

Figure 5. Example CNN Architecture [9]

TABLE II. RESULTS FOR TRAINING PRE-PROCESSED DATASET WITH SIMPLE CNN ARCHITECTURE [9]

Parameters	**Number of epochs**	**Training Accuracy**	**Validation Accuracy**
Validation split = 0.2, RGB image dataset	5	97.78 %	55.56 %
	10	98.33 %	54.44 %
	15	99.72 %	56.67 %

The pre-processed dataset was also trained using AlexNet and the results are shown in Table III. In this training model, the validation split (percentage of training images selected for validation) was 0.2 and the learning rate was set to 0.0001. As seen in all the result tables, the AlexNet architecture provides the maximum validation accuracy of all the architectures and corresponding options attempted. Once training of the AlexNet neural network is complete, the model can be exported and deployed for the identification of various unknown plant images. Training the AlexNet architecture on images of animals or insects could additionally allow it to identify images of animals or insects.

TABLE III. RESULTS FOR TRAINING PRE-PROCESSED DATASET WITH ALEXNET [10]

Parameters	**Number of epochs**	**Training Accuracy**	**Validation Accuracy**
Validation split = 0.2, RGB image dataset	5	20.00 %	20.00 %
	10	56.11 %	58.89 %
	15	78.61 %	72.22 %

VI. CONCLUSION

With the proposed artificial intelligence-oriented solution for plant identification, engineers can design simple, portable, and cost-efficient devices for plant species identification. The experiments were limited by the data available by LeafSnap, the computation power of Jetson Nano as well as the strength of the algorithm. Attempting to achieve high accuracy on the validation dataset when only provided with a limited amount of data proved to be a challenging task. Since AlexNet produced the best results, it is recommended to use AlexNet over any variation of the simple CNN architecture to achieve maximum recognition accuracy.

REFERENCES

[1] H. Yalcin and S. Razavi, "Plant classification using convolutional neural networks," *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, Tianjin, pp. 1-5, 2016.
[2] Swedish Leaf Dataset: https://www.cvl.isy.liu.se/en/research/datasets/swedish-leaf/
[3] Flavia dataset: http://flavia.sourceforge.net/
[4] LeafSnap Dataset: http://leafsnap.com/dataset/
[5] NVIDIA Jetson Nano: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[6] Varghese, Baizel Kurian, Albin Augustine, Jubil Maria Babu, Deepa Sunny, and Er Sijo Cherian. "INFOPLANT: Plant Recognition using Convolutional Neural Networks." In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 800-807. IEEE, 2020.

[7] Choi Sungbin. "Plant Identification with Deep Convolutional Neural Network: SNUMedinfo at LifeCLEF Plant Identification Task 2015." In *CLEF (Working Notes)*. 2015.

[8] Xiao, Qingguo, Guangyao Li, Li Xie, and Qiaochuan Chen. "Real-world plant species identification based on deep convolutional neural networks and visual attention." *Ecological Informatics* 48 (2018): 117-124.

[9] Image Classification using TensorFlow: https://www.tensorflow.org/tutorials/images/classification

[10] AlexNet: https://en.wikipedia.org/wiki/AlexNet