Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 44 (2023), No. 4, pp. 709–716
DOI : 10.47974/JIOS-1269

Deep learning for automatic identification of plants through leaf

Silky Sachar *
Anuj Kumar †
Department of Computer Science and Applications
Panjab University
Chandigarh
India

Abstract

Automatic identification of plants, has been a widely explored field for the conservation of environment. Deep Learning has been extensively used in image recognition tasks due to its powerful ability to extract features from the given set of images. In this paper, we have trained Convolutional neural Network models from scratch by first pre-processing the images using MobileNet's pre-processing input function to identify the plant species using leaf images. Four CNN models are discussed at different depths to understand how the accuracy of identification can be improved and the impact of hyperparameters namely batch size and number of epochs have on the accuracy of identification. The four models have been evaluated on two freely available leaf datasets: Flavia and Swedish. To reduce overfitting, data-augmentation and Early Stopping callback has been applied. The performance of the proposed CNN model was also compared to SVM, Random Forest and K-Nearest Neighbors classifiers on both datasets. Maximum accuracies were reported to be 95.35 % and 95.24% on Flavia and Swedish respectively.

Subject Classification: *68T07.*

Keywords: *Deep learning, Machine learning, MobileNet, Convolutional neural networks, Pattern recognition, Computer vision.*

I. Introduction

Human beings have been utilizing plants for all their essential needs [1] but most species are at the verge of extinction, which raises a need to develop an automated plant identification system. Traditional

* *E-mail:* **Silky.sachar@gmail.com (Corresponding Author)**
† *E-mail:* **Anuj_gupta@pu.ac.in**

methods were time consuming and very complex. Advances in the field of computer vision show how effectively the machine Learning techniques can be applied for plant identification at a faster rate, with a better accuracy and without the need of expert intervention. In literature, there have been attempts to identify plants using their flowers, stems, fruits, bark but the most abundantly available plant part, which makes their identification easier is the Leaf [2]. The leaf image can be captured by using a camera or a cell phone to be input into the system and its name and species can be acquired. Different features of a leaf include its shape, texture, color and venation patterns. Shape of the leaf include the region based morphological and digital features and contour based features [3]. Texture feature extraction based on grayscale images has been obtained using Modified Local Binary Patternand wavelets.

Recently, it has been observed that the most sought-after technique in the field of image recognition is deep learning. It has found applications in various biological, medical, computer vision and other tasks [4-8].

In this research, we preprocess the augmented input images using MobileNet's preprocess input function and then develop and train four Convolutional Neural Networks (CNN) of increasing number of layers. After exhaustive experimentation, the final proposed model consists of 5 conv. layers (4 with 64 filters and 1 with 128 filters) +1 softmax. The training, validation and testing accuracy metrics for different hyperparameters (number of epochs and batch size) was observed. The performance of deep learning was also tested against the performance of Machine Learning classifiers namely Support vector machines (SVM), Random Forest and K-nearest neighbors(k-NN).

The paper is divided as follows: Section II consists of recent studies in identification of plants from leaf images. Section III describes the methodology and Section IV presents the results of the experiments. Section V concludes the paper and presents future scope in this field.

II. Related Studies

Pattern recognition with Neural Networks has been discussed in detail in and it has only been evolved in the last decade. The application of Probabilistic Neural Network was observed in to accurately classify 1800 leaves into 32 classes utilizing 12 features. Deep learning has proven to be more robust and powerful thus eliminating the need of hand-crafted feature extraction methods. An excellent example where the leaf vein patterns were used to classify legumes into 3 different classes was presented in where the authors developed 6 different models incrementing the number of layers to observe the accuracy of the model. They conclude that the use of deep learning CNN approaches provides an efficient feature extracting and classification methods. Authors in developed a CNN model from scratch and tested it against the dataset of 4 classes manually collected by them.

III. Materials and Methods

A. *Experimental Setup*

The training and testing of the deep CNN models were implemented using scikit-learn Keras and tensorflow framework which use python programming language. The training and testing were executed on laptop with 16GB RAM, using NVIDIA GeForce RTX 3070, GDDR6 @ 8GB memory.

B. *Datasets and Pre-processing*

The experiments were conducted on freely available leaf image datasets namely Flavia and Swedish. Flavia consists of 32 species of leaves of common plants found in Yangtse Delta in China. Each class has 50-60 images. Swedish dataset consists of 15 species of leaves found between the Linkoping University and the Swedish Museum of National History. Each class consists of 75 images. The images were divided into training set, validation set and testing set.

The number of images per class were increased in training set of both cases by applying augmentation as a greater number of samples allow the network to gain better knowledge about the features of images. This technique is also important as it helps the model to prevent overfitting. While the images were loaded for training, then again real-time augmentation was applied using Keras ImageDataGenerator. These transformations included: featurewise_centre, featurewise_std_normalization, rotation_range=20, width_shift_range=0.2, height_shift_range=0.2, horizontal_flip, shear_range=0.2, zoom_range=0.2. Real-time augmentation is not to increase the number of images but to apply the above-mentioned transformations in each epoch and then go for training.

C. *Architecture of Convolutional Neural Network*

Deep Learning is a sub-class of Artificial Intelligence which has been used for the automatic recognition of objects from images.

Convolutional Neural networks (ConvNet) consist of multiple hidden layers that perform feature extraction from an image. The basis of ConvNet is the convolution layer which extracts high level features from the images. The number of filters (64 and 128) used in the models perform the convolution operation. The filter matrix of size 3x3 is slided over the image matrix to compute the dot product to finally obtain the convolved feature matrix. Padding is the process of adding pixels to the edge of the image so as not to lose the bordering image information. Padding is set to "Same" which calculates and add the required padding to the image to ensure that output image has the same size as the input image. The stride value sets the amount of movement between the application of filter and the input image. The stride value is set to 2 meaning that filter moves 2 pixels right for each horizontal movement and 2 pixels down for each vertical movement. The activation function used here is Rectified Linear Unit (Eq.1) which can be places at the end or in between Convolution layers. It ensures that the output value always lies between 0 and +infinity.

$$\text{output} = \max(0, \text{input}) \qquad \text{...(Eq.1)}$$

This means that negative values will be 0 and output will be Identity for all positive values.

Normally we do not want to take all values from the input into the next layer, but by specifying "MaxPooling" we will only take the Maximum summarized value of all the values present in the input. The Flatten layer then converts the output of the convolution layer into a 1D feature vector which can be input into the fully connected Dense layer. Softmax activation function (Eq.2) acts as a classifier and converts the feature vector into vector of probabilities, where the probability of each value is proportional to the relative scale of each value in the feature vector.

$$\sigma(\vec{z})i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad \text{...(Eq. 2)}$$

Where, the input vector Z_i constitutes values from (z_0 to z_k) which can take up any real value, which may or may not pose a valid probability distribution, due to which softmax operation would be necessary.

e^z : the exponential function is applied on the input values.

$\Sigma_{j=1}^{K} e^z$: this is responsible for normalization of the output values so that all values lie in the range 0 and 1.

K : this is the number of classes in the multi-class classifier.

D. *Construction of Models*

Both datasets were first augmented to increase the number of images to about 160 in each class and then while loading the RGB images, real time augmentation was applied and images were resized into 224x224. The images were then preprocessed using pre-processing function used while developing the MobileNet pretrained CNN model. Four CNN models of varying depth were developed and tested on both datasets Flavia and Swedish. Adam Optimizer with learning rate of 0.0001 and categorical_crossentropy loss function.

Four CNN models of increasing were developed from scratch. These models are called M1, M2, M3 and M4. M1 consists of 2 Conv layers (filters 64) + 1 softmax. M2 consists of 3 Conv layers (2 with filters 64 and 1 with filters 128) + 1 softmax. M3 consists of 4 Conv layers (2 with 64 filters and 2 with 128 filters) +1 softmax. M4 consists of 5 conv layers (4 with 64 filters and 1 with 128 filters) +1 softmax. All conv layers are followed by "relu" activation and MaxPooling2D layers. The selection of filters was done using kerastuner and then the best combination of layers and filters were used in the experiment. Batch size is an important hyperparameter used in development of CNN models. It refers to the number of training examples

Table 1

Training, validation and Testing accuracies of all models(M1,M2,M3,M4) on Flavia Dataset

Model Name	Batch Size	Number of Epochs	Training accuracy	Validation Accuracy	Testing Accuracy
M1	16	60	98.21	78.09	77.67
	32	8	98.01	86.51	86.51
	64	17	97.86	85.96	80.93
M2	16	81	98.8	94.94	94.88
	32	50	97.68	91.01	92.09
	64	67	98.56	86.52	86.52
M3	16	14	98.75	92.7	93.02
	32	16	99.01	92.13	92.13
	64	54	98.07	83.15	84.65
M4	16	22	97.74	94.94	93.95
	32	54	97.12	91.57	92.09
	64	78	97.87	94.94	95.35

Table 2

Training, validation and Testing accuracies of all models(M1,M2,M3,M4) on Swedish Dataset

Model Name	Batch Size	Number of Epochs	Training accuracy	Validation Accuracy	Testing Accuracy
M1	16	34	97.96	78.1	**84.17**
	32	45	95.56	72.38	81.67
	64	54	96.37	78.1	75.83
M2	16	27	97.96	89.52	90.83
	32	45	96.45	80	79.17
	64	54	97.64	88.57	**94.17**
M3	16	09	95.95	87.62	89.17
	32	39	95.9	92.38	94.17
	64	47	97.47	88.57	**94.17**
M4	16	15	97.55	**95.24**	92.5
	32	18	97.55	92.38	**93.33**
	64	58	96.34	83.81	91.67

used in 1 iteration. In our experiments, we have used epoch size 16, 32 and 64. One Epoch refers to one pass over the entire dataset, separates entire training into different phases. Number of epochs in these experiments is set to 100, but the Early Stopping callback is applied to ensure that training stops after there is no improvement in validation loss over 5 consecutive epochs.

IV. Results and Discussion

Table 1 and Table 2 show the training, validation and testing accuracies obtained for the models M1, M2, M3, M4 on Datasets Flavia and Swedish respectively. Maximum validation and testing accuracy on Flavia dataset was 94.94% and 95.35% and on Swedish dataset it was 95.24% and 94.17% respectively. It can be observed from the table 1 that on the Flavia dataset with 32 classes, the Adam optimizer with learning rate 0.0001 correct allignment shows different behavior for shallow models M1, M2 where the smaller number of batch size trains for more number of epochs and as the depth is increased for models M3 and M4 less number of samples in batch size trains for a smaller number of epochs.

From Table 2 it is also observed that on the Swedish Dataset which has 15 classes, the behaviour of optimizer is similar across all four depths, that is for lower batch size, the training goes on for a smaller number of epochs.

It is observed that the model M4 shows the maximum validation accuracy of 95.24% and models M2 and M3 show more testing accuracy than model M4 which also lays stress on the fact that along with the depth, the width of the CNN model also plays a vital role in the accurate classification of the images into their respective classes.

Further, in order to compare the performance of the deep Convolution Neural Networks to the traditional Machine Learning classifiers, the augmented image datasets were pre-processed using the pre-process function from the MobileNet pre-trained model. These images were then input into the SVM, Random Forest and k-NN classifiers. Out of the 3 classifiers, k-NN performed the best in both datasets. The testing accuracies were compared to the proposed CNN model M4 and the results showed that the deep learning model outperforms the traditional Machine Learning classifiers

References

[1] A. Antonelli *et al.*, "State of the World's Plants and Fungi 2020," p. 100, 2020, doi: doi.org/10.34885/172.

[2] J. Wäldchen and P. Mäder, "Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review," *Arch. Comput. Methods Eng.*, vol. 25, no. 2, pp. 507–543, Apr. 2018, doi: 10.1007/s11831-016-9206-z.

[3] S. Sachar and A. Kumar, "Survey of feature extraction and classification techniques to identify plant through leaves," *Expert Syst. Appl.*, no. October, p. 114181, 2020, doi: 10.1016/j.eswa.2020.114181.

[4] H. Zhang and X. Tao, "Leaf image recognition based on wavelet and fractal dimension," *J. Comput. Inf. Syst.*, vol. 11, no. 1, pp. 141–148, 2015, doi: 10.12733/jcis12815.

[5] A. Dureja and P. Pahwa, "Medical image retrieval for detecting pneumonia using binary classification with deep convolutional neural networks," *J. Inf. Optim. Sci.*, vol. 41, no. 6, pp. 1419–1431, 2020, doi: 10.1080/02522667.2020.1809096.

[6] B. M. Joshi and H. Bhavsar, "Plant leaf disease detection and control: A survey," *J. Inf. Optim. Sci.*, vol. 41, no. 2, pp. 475–487, 2020, doi: 10.1080/02522667.2020.1734295.

[7] D. Virmani, C. Gupta, P. Bamdev, and P. Jain, "iSeePlus : A cost effective smart assistance archetype based on deep learning model for visually impaired," *J. Inf. Optim. Sci.*, vol. 41, no. 7, pp. 1741–1756, 2020, doi: 10.1080/02522667.2020.1812198.

[8] V. Gupta, S. Juyal, G. P. Singh, C. Killa, and N. Gupta, "Emotion recognition of audio/speech data using deep learning approaches," *J. Inf. Optim. Sci.*, vol. 41, no. 6, pp. 1309–1317, 2020, doi: 10.1080/02522667.2020.1809089.

Received September, 2022