Journal of
Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.755153

PLANT IDENTIFICATION AND CLASSIFICATION THROUGH CONVOLUTIONAL NEURAL NETWORKS: A DEEP LEARNING APPROACH

Dr.V.Vaidehi[1], Dr. Raynukaazhakarsamy[2]
[1]Assistant Professor, Faculty of Computer Applications, Dr.M.G.R. Educational and Research Institute, Chennai, Email: vaidehi.mca@drmgrdu.ac.in
Assistant Professor(SG-I), Department of Computer Applications,
[2]Nehru Arts and Science College (Autonomous), Nehru Gardens, Thirumalayampalayam, Coimbatore - 641 105
Email:rayahsoct1918@gmail.com

Abstract
Six commonly found native medicinal herbs from Tamil Nadu make up the dataset. For primary classification tasks, a powerful CNN (Convolutional Neural Network) model is used. Thirty percent of the dataset is set aside for validation and the remaining seventy percent is used to train the model. Techniques for dataset augmentation and randomization are used before model input. Using SVM (Support Vector Machine) classifiers in conjunction with a one-versus-all coding architecture, the ECOC (Error Correcting Output Codes) framework is put into practice. CNN models are used to extract features, which are then fed into the classification model.
Keywords: CNN, SVM, Deep Learning, ANN (Artificial Neural Network), Classification, Performance Evaluation

1. INTRODUCTION

This paper explores the fields of computer vision and image processing in great detail, demonstrating the usefulness of deep learning and machine learning techniques for a range of processing jobs. It particularly highlights how important these cutting-edge technologies are to the identification and classification of various plant species. The chapter also discusses the most recent developments and crucial computer vision tools that are relevant to plant identification.

The results of class labels are obtained by grouping. Using a dataset of 1335 photos, the experiment achieved a remarkable accuracy rate ranging from 89% to 93%. SVM is used for classification along with the softmax function. In SVM-based problem formulations, the One vs. All (OVA) approach becomes a popular method for multi-classification tasks.

2. PLANT IDENTIFICATION SURVEY

In order to classify plant species, Dyrmann and Midtiby (2016) used CNN on a total of 10,413 pictures that represented 22 crop and weed species at different growth phases. For training and evaluation purposes, the writers took vertical photos of seedlings from 22 distinct plant species. With an average accuracy of 86%, the network's categorization accuracy ranged from 33% to 98%.

To identify plant species, Ghazi, Yanikoglu, and Aptoula (2017) used several deep convolutional neural networks. They conducted a comparative study of popular CNN designs, including VGGNet, AlexNet, and GoogleNet. Different plant elements, such as branches, whole plants, flowers, fruits, leaves, and stems, were used to train these networks. The best results were obtained by fine-tuning VGGNet, which led to an overall accuracy of 78.44%.

In 2012, Gopal, Reddy, and Gayatri suggested an automated technique for leaf identification of particular medicinal plants. To differentiate between various leaf kinds, their method made use of boundary-based characteristics, moment features, and color features. Training was done on one hundred leaves, and another fifty leaves were used for evaluation. The system attained an impressive 92% categorization efficiency.

An Artificial Neural Network (ANN)-based model was created by Janani and Gopal (2013) to categorize different kinds of medicinal plants according to their leaf features. The model, which examined eight crucial characteristics—compactness, eccentricity, skewness, kurtosis, energy, correlation, sum-variance, and entropy—achieved an accuracy rating of 94.4% using a dataset of 63 leaves.

In order to identify knives in photographs, Kibria and Hasan (2017) looked into the Bag of Words, HOG-SVM, CNN, and Pre-trained AlexNet algorithms. They discovered that the greatest results for detection and classification came from using SVM with pre-trained AlexNet.

A technique for identifying vehicles by reading their number plates was demonstrated by Shima (2016). The method extracted features using a pre-trained Convolutional Neural Network (CNN) and employed a Support Vector Machine (SVM) classifier. On a dataset of 126 sample photos, the approach effectively detected number plate regions at different distances in rear-end images with an accuracy of 89.7%.

3. PROPOSED METHODOLOGY

The following flow diagram shows how this study has been organized into two separate models:

Model 3.1: CNN Architecture-Based Accuracy Level Validation Model

3.2: CNN Architecture-Based Improvement of Data Augmentation

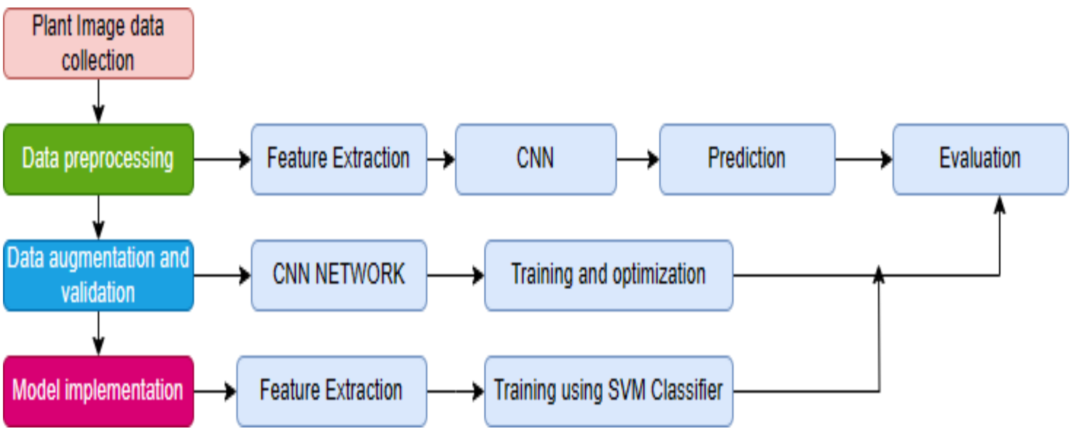


FIG1.1:OVERALL FLOWDIAGRAM OF PLANT IMAGE CLASSIFICATION

Table 1.1: Plant Dataset

	AVARAI	FENUGREEK	GUAVA	NAVAL	NEEM	TULSI
No.ofimages	600	600	600	600	600	600

This study's methodology entails building a model from the ground up and then training it. Data annotation, which includes text, photographs, and videos used to identify or annotate things of interest within images, is done before training. By ensuring precision, this method makes it possible for computers to use computer vision algorithms to recognize these items.

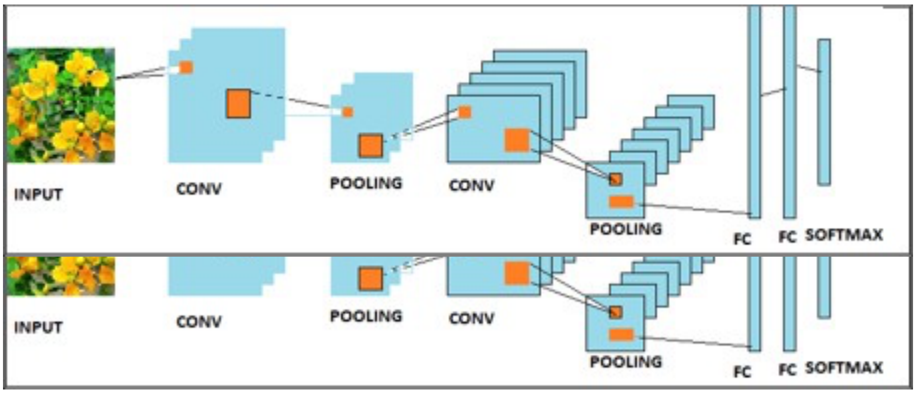
All class data needs to be kept organized and inside the appropriate folder. To do this, a distinct folder must be made for every class or category that is being considered. We then go ahead and build our neural network. Convolutional Neural Networks (CNNs) are the go-to option for working with picture data. As a result, we will create a CNN and train it on the dataset of plant images. To reduce overfitting, a convolutional layer is added first, then a pooling layer and, if desired, a dropout layer. Dense fully connected layers are appended at the end. The number of units in this layer corresponds to the number of classes we hope to forecast, and the final layer outputs the predictions or findings.

Model 3.1:Accuracy level confirmation through CNN architecture

Figure 1.3 shows the architecture of a convolutional neural network (CNN). A CNN is made up of multiple layers that are intended to anticipate and extract information from input data. Convolutional, pooling, and fully connected layers are some of the layers that collaborate to do this task. Convolutional layers use a sequence of trainable filters to analyze the input data, preserving spatial linkages while capturing local patterns and characteristics. In order to extract dominating characteristics while lowering spatial dimensions, pooling layers downsample the feature maps. For effective and reliable feature representation in the network, this downsampling makes sure that crucial information is kept intact.

Fully connected layers integrate local information from the convolutional layers to enable high-level feature learning and classification. They accomplish this by establishing connections

Fig1.3: CNN Architectures and their layers



between each neuron in the next layer and every neuron in the layer before it. First, we will provide some essential parameters to configure our CNN. We'll use a dataset of 2,000 sample photos and set the batch size to 20. There will be 100 iterations in each epoch. We will use a validation set of 1,000 photos to assess the model during the course of our 30-epoch training procedure.

We are going to create a simple CNN model with three convolutional layers and add max-pooling layers in order to automate the process of extracting features from our photos. Dropout layers will also be included in order to reduce the possibility of overfitting. Finally, a dense, fully linked layer that produces predictions for six classes will complete the model.

If the model stops getting better after five consecutive epochs, we will use early stopping to end the training process. By ending training early, when the model no longer improves, overfitting is avoided. The training procedure will end if validation performance does not improve after five epochs, as indicated by the parameter patience=5.

PERFORMANCE EVALUATION

Confusion Matrix

One often used statistic to assess a classification system's performance is the confusion matrix. It is a performance metric for classification issues in machine learning involving two or more output classes.

Precision

By dividing the number of accurate predictions by the total number of data points in the dataset, accuracy, or ACC, is calculated. It has a range of 0.0 to 1.0, with 1.0 denoting ideal precision. An alternative method of calculating accuracy is to deduct the error rate (ERR) from 1.

Sensitivity (True Positive Rate or Recall)

The number of accurate positive predictions divided by the total of accurate positive predictions and false negatives yields the sensitivity (also known as recall or true positive rate). It has a range of 0.0 to 1.0, where 1.0 is the highest sensitivity level and 0.0 is the lowest.

The True Negative Rate of Specificity

The number of accurate negative predictions divided by the total of accurate negative predictions and false positives yields the specificity (also known as the true negative rate). Its values fall between 0.0 and 1.0, with 1.0 denoting the optimal specificity and 0.0 denoting the poorest.

Rate of False Positives

By dividing the total number of false positives by the sum of the true negatives and false positives, the False Positive Rate (FPR) is determined. It has a range of 0.0 to 1.0, where 0.0 is the best rate and 1.0 is the worst. Furthermore, FPR can be computed as 1 - specificity.

Rate of Misclassification (MCR)

The calculation of the Misclassification Rate (MCR), which is often referred to as classification error, involves dividing the total number of data points by the sum of false positives and false negatives. It is also possible to calculate it as 1 minus precision.

Table1.2: CNN Performance metrics from scratch

Class	truth overall	classification overall	ACC	PREC	REC	F1	TP	TN	FP	FN	ERR	TNR
1	83	94	92.93	0.74	0.84	0.79	70	424.9	15	25.4	0.08	0.97
2	90	71	92.93	0.87	0.69	0.77	62.7	433.4	29.6	10.4	0.07	0.94
3	88	97	93.31	0.77	0.85	0.81	75	422.9	14.4	23.8	0.07	0.97
4	70	92	93.88	0.71	0.93	0.8	65	435.7	6.2	29.2	0.07	0.99
5	87	79	91.22	0.76	0.69	0.72	60.2	425.9	29.2	20.8	0.09	0.94
6	105	90	89.87	0.79	0.68	0.73	71.8	405.5	37	21.8	0.11	0.92

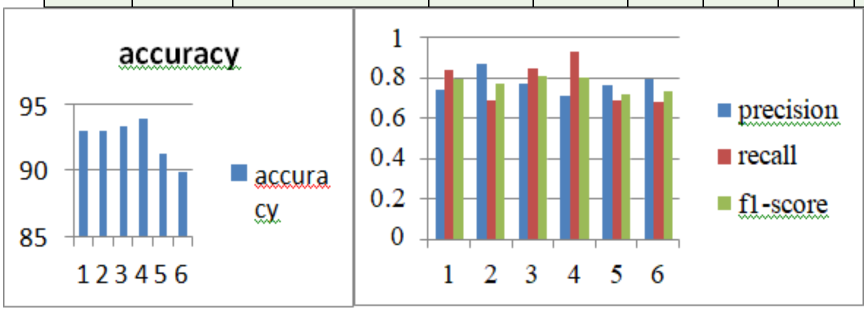


Fig1.4:CNN Performance metrics chart

Overall accuracy: 77.05%

The average accuracy in our validation set is about 77%. When there is insufficient training data and the model sees the same patterns repeatedly over time and epochs, this is known as overfitting. By adding variations of the original photos to our current training data, image augmentation can help reduce overfitting.

Model 3.2:Data Augmentation with CNN architecture

Image augmentation is the process of creating synthetic images in memory using operations like rotation, shifting, and zooming. Rotation entails turning the picture at different angles. Shifting causes "holes" in the visual pattern that need to be filled in by interpolating either vertically or horizontally. Among other things, zooming produces an enlarged version of a certain area of the original image.

The Image Data Generator tool allows us to apply various transformations, including shear, rotation, zoom, width shift, height shift, and horizontal flip, to generate synthetic images. Once these procedures are completed, new pixels can be filled in using the `fill_mode` argument. The validation accuracy rises to about 80% when CNN is used in conjunction with picture augmentation, which is an improvement over our prior model. Furthermore, our model is no longer overfitting, as the validation accuracy is really near to the training accuracy.

Table1.3:CNN Performance metrics after augmentation

Class	truth overall	classification overall	ACC	PREC	REC	F1	TP	TN	FP	FN	ERR	TNR
1	87	97	96.58	0.85	0.94	0.89	82.7	408.5	17.2	6.1	0.05	0.96
2	102	97	94.01	0.85	0.8	0.82	82.7	476	17.2	23.3	0.07	0.97
3	93	98	94.69	0.82	0.86	0.84	80	484.7	19.7	14.8	0.06	0.96
4	95	98	94.01	0.81	0.83	0.82	79	482	20.9	17.3	0.06	0.96
5	103	97	92.81	0.81	0.77	0.79	79	401.9	20.9	27	0.09	0.95
6	104	97	93.66	0.85	0.79	0.82	82.7	474.7	17.2	24.6	0.07	0.97

OverallAccuracy:82.88

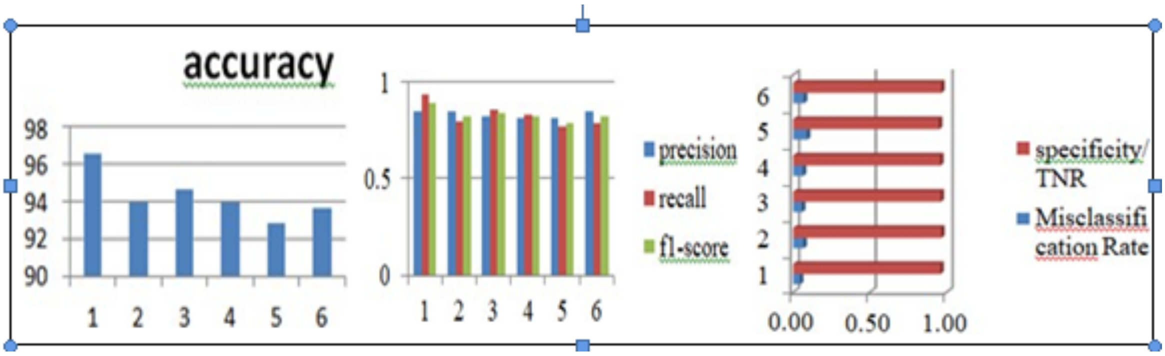


Fig1.5:CNN Performance metrics chart after augmentation

4. CONCLUSION&FUTUREWORK

In order to safeguard the environment, progress agriculture, conduct medical research, and provide food, plants are essential to society's development. But classifying plant species, determining their diseases, and assessing plant yield are getting harder and harder. Two models were suggested in this study for the classification of image categories, and we came to the conclusion that the best model fit the available dataset.

Model 1 began with two thousand samples in total. After two to three epochs, the model started to overfit the training set. It used three convolutional layers to extract features and a flatten layer to handle the feature maps that were produced. Model 1 had an accuracy rate of 77.05%

on average. After a few epochs, Model 1 overfitted due to the small number of data samples. We used Data Augmentation using a CNN architecture (Model 2) to increase performance. By applying picture alterations, this technique generated different copies of the same images each time, increasing the amount of training data. The accuracy average for Model 2 was 82.05%.

The validation accuracy increased to almost 80% with CNN and picture augmentation, outperforming the previous model's output. This model's similarity to training and validation accuracy indicates that it is no longer overfit.
The findings demonstrate that our suggested model, which combines CNN architecture with data augmentation, performs better than the standard CNN model in terms of testing and training accuracy, attaining an 80% validation accuracy. As a result, the CNN architecture model that uses data augmentation is the best model in this investigation.
Subsequent research endeavors will center over augmenting efficacy through pre-trained CNN architectures and refining them through transfer learning utilizing the dataset. The scope of this endeavor will also include using deep learning techniques to identify plant diseases. Our goal is to improve the accuracy of plant disease diagnosis by identifying diseased parts' features and categorizing the target disease locations.

References

1) Dyrmann, M, Henrik, K & Midtiby, HS 2016, ‘Plant species classification using deepconvolutional neural network’, *Biosystems Engineering,* vol. 151, no. 1, pp. 72-80

2) Ghazi, MM, Yanikoglu, B & Aptoula, E 2017, ‘Plant identification using deep neuralnetworks via optimization of transfer learning parameters’, *Neurocomputing,* vol. 235,no. 1, pp 228-235

3) Gopal, A, Reddy, SP & Gayatri, V 2012, in proceedings of International ConferenceOn Machine Vision and Image Processing, December 14 - 15, ‘Classificationofselectedmedicinal plants leaf using image processing’, Coimbatore, India

4) Janani, R & Gopal, A 2013, in proceedings of International Conference on Advanced Electronic Systems, September 21 - 23, ‘Identification of selected medicinalplant leaves using image features and ANN’, Pilani, India

5) Kibria, SB & Hasan, MS 2017, in proceedings of 2ndInternational Conference onElectrical & Electronic Engineering, December 27 - 29, ‘An analysis of Featureextraction and Classification Algorithms for Dangerous Object Detection’, Rashahi, Bangladesh

6) Shima,Y2016,inproceedingsofIEEEInternationalSymposium onRoboticsandIntelligentSensors (IRIS) IEEE, December 17 - 20, ‘Extraction of number plate images based on image category classification using deep learning’, Tokyo, Japan