CROP CLASSIFICATION USING DEEP LEARNING

A Project Report

Submitted by

AADHITH S. [CB.EN.U4CSE18101]
HAREESH V. [CB.EN.U4CSE18118]
VIGNESH H. [CB.EN.U4CSE18163]
THARUN KUMAR A. [CB.EN.U4CSE18173]

Under the guidance of

Dr. S. Padmavathi, M.E, Ph.D

(Associate Professor(SG), Department of Computer Sciene & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

Amrita Nagar PO, Coimbatore - 641 112, Tamilnadu

November 2021

ABSTRACT

The modern lifestyle has made people less aware of where we get our food from. From a commoner's perspective, it may seem irrelevant but it is our responsibility and is necessary knowledge to know where, when, and how our food is being produced. So, we are designing a web-based application where a users can upload images and gain insights on the crop.

The workflow is as follows, to decide a few couples of crops whose images are abundantly available, to build a deep learning model, to train the same with the data sets, also train and test various other machine learning architectures, compare the best among all and select the best classification model to classify crops and finally build a user interface to integrate the model and make an end product.

We expect this project to give the users more affinity towards farming and pave a path to bring awareness on agriculture as a profession.

TABLE OF CONTENTS

ABBREVIATIONS

CNN Convolutional Neural Network

ML Machine Learning

DL Deep Learning

MSE Mean Square Error

DN Deconvolutional Networks

UI User Interface

Chapter 1

INTRODUCTION

1.1 Problem Definition

Develop a web application that can identify crops using the image of it's leaf with the help of a Machine Learning (ML) or Deep Learning (DL) model. The application will allow users to either upload or take new image of the leaf.

1.2 Motivation and need

The modern lifestyle has made people less aware of where we get our food from. From a commoner's perspective, it may seem irrelevant but it is our responsibility and a necessary knowledge to know where, how and when our food is being produced. So we are designing a web-based application where a user can upload an image/images to gain insight into the crop.

Chapter 2

LITERATURE SURVEY

2.1 Classification of plant leaf images with complicated background

Authors: Xiao-Feng Wang, De-Shuang Huang, Ji-Xiang Du, Huan Xu, Laurent Heutte (Wang et al., 2008)The crux of the research is to identify different plants based on the image of its leaf. The image can have any unnecessary background objects like stones, dead leaf and even an overlapping of another leaf provided at least one leaf is fully visible in the picture. The technique of automatic marker-controlled watershed segmentation is used. The first being watershed segmentation, which provides a grayscale image with the target leaf with other leaves and interferents like stone. This image is cleaned by using Otsu thresholding which in turn provides markers which defines the boundary of the image to be included. An erosion operation is done over the image to remove the leaf stalk and other irrelevant details to get the exact image of the shape of the leaf alone. From this image a pattern is obtained, i.e., a feature vector that includes 7 Hu geometric moments and 16 Zernike moment. These features are now passed to a Moving Center Hypersphere (MCH) classifier having the idea that each class of patterns in n-dimensions can be represented by using a series of n-hyperspheres. The classifier provides result with an accuracy of 92.6% on a 1200 leaf sample of 20 classes of plants.

2.2 Mobile plant leaf identification using smart-phones

Authors: Bin Wang, Douglas Brown, Yongsheng Gao, John La Salle
(Wang et al., 2013)The authors have created new mathematical formulae, a novel multiscale shape descriptor, to identify the plant species based on the image of a single leaf. The leaf shape extractor has two stages. The first being a shape descriptor, a mathematical coordinate function normalized by the perimeter of the contour. The second

being a dissimilarity measure. A comparison between this method and others is done by means of “bulls-eye test”. The authors implemented a mobile application coded in Java as the User Interface (UI), which sends the image to a server which does the required operations to identify the leaf. This proposed novel multiscale shape descriptor achieves a higher effectiveness and superior efficiency over methods like Fourier descriptor, Multiscale Convexity/Concavity representation (MCC), Triangle-Area Representation (TAR), Inner Distance Classifier (IDSC) at an accuracy rate of 96.05% on a dataset of 1200 leaves of 100 plant species.

2.3 Leaf Plant Identification System On Hidden Naive Bayes Classifier

Authors: Heba F. Eid, Aboul Ella Hassanien

(Eid et al., 2015)A Plant identification model based on the plant’s leaf shape and vein pattern. In this model Hidden naive Bayes classifier is used to classify the plant. This plant classification model takes into account two types of features for classification, the morphological vein features, and the shape geometric features. Where 10 features are exacted and passed to the hidden naive classifier. The vein features are extracted based on gray-scale morphology. While, the shape geometric features are: Leaf Area, Leaf Perimeter, Leaf Diameter, Leaf Length, Leaf Width, Covex area. To extract the feature first the RGB leaf image is converted into a grayscale image. Step 2 converting Gray Scale Image into a binary image and Noise filter by Laplacian filter of 3X3 spatial mask then finally binary image transformers to its inverse image next vein features are obtained by performing the morphological operation on the grayscale image. The purpose of applying morphology processing on a grayscale image is to get rid of the gray overlap between the leaf vein and the background. The features are then passed to hidden naive Bayes for classification. Results of Hidden Naïve Bayes classification are Precision of 91.9% for classification based on plant leaf shape only, 73.5% for classification based on plant vein features only, and 97.0% for combining these both features.

2.4 Plant Identification with Convolutional Neural Networks

Authors: Sue Han Lee, Chee Seng Chan, Paul Wilkin, Paolo Remagino

(Lee et al., 2015)In this paper, They have employed deep learning in a bottom-up and top-down manner for plant identification. In the former, they have chosen to use a convolutional neural networks model to learn the leaf features as a means to perform plant classification. In the latter, rather than using CNN as a black box mechanism, where they employ Deconvolutional Networks (DN) to visualize the learned features. The CNN model has a precision of 98.1% which outperforms state-of-the-art solutions that are employed carefully chosen hand-crafted features even when different classifiers are used. The classification accuracy of the CNN model trained using D2(feature) with Multilayer perceptron is the highest. From these results we can justify that learning features through CNN can provide better features.

2.5 Plant Recognition using Convolutional Neural Networks

Authors: Baizel Kurian Varghese, Albin Augustine, Jubil Maria Babu, Deepa Sunny, Er.Sijo Cherian

(Varghese et al., 2020)This literature survey by the CSE students of Saintgits College of Engineering, discusses the recognition of plants using the Convolutional Neural Network. They have identified MobileNet Convolutional Network to work with the problem. Trained images are stored in a .tflite file. The labels and other information regarding the plants are stored in the database with the label. Image of a leaf is given as an input and feature extraction is done using the Convolutional Network that consists of many layers like convolutional layers, ReLU layer, pooling layer and finally the fully connected layer. Each layer has its own purpose where the Convolutional layers convert the original image of higher dimensions to lower by convolving using the filters. The ReLU is an activation layer where the pixel values are converted as follows: f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero.

Pooling layer reduces the dimension of the feature map, downsample each feature map independently. Finally in the fully connected layer flattening is done to make 2D data into 1D data vector and finally the classification is done. The new image is checked and predicted with the .tflite file and checked with the labels. The information stored in the database is then fetched and diplayed to the user.

2.6 Image Processing Methods for Identifying Species of Plants

Authors: Shulin Dave, Ken Runtz

(Dave and Runtz, Dave and Runtz)The objective of this literature survey was to apply 3 different image processing techniques to identify plants. The input images were scanned and converted from TIFF image format to binary RGB 256 x 256 x 8 bits. A black and white image was generated from RGB format image file. Both colour and black and white images were used in different image processing techniques.

Frequency Domain Method:

Here the digital image is being converted to frequency domain from spatial domain. Fourier transformation is the tool that is used for this process. After transformation, power spectrum inside the annular band of radius R1 to R2 is calculated. The centre of the image is the centre point of the annular band. This user defined band keeps on sliding towards the edge of the image. A plot of power capsulated in the annular band versus radius is plotted. The relative frequency is observed. This relative frequency forms the basis of identification plants.

Fractal Dimension Method:

Used in classification of natural shapes and textures. Fractal dimension can be calculated by $D = \log(N) / \log(r)$, we can calculate the dimension D by seeing how the number of units, N, changes with the magnification factor, r. The fractal dimension D is computed from the black and white plant image. By comparing the fractal dimension of different varieties of plants the program identifies the type of plant in the image.

Co-occurrence Matrix Method:

Small areas of image from the pre-processed black and white images are taken and processed independently by computer. Texture and tone are the most important. Gray tone Spatial-Dependence Matrices are calculated from the image. All the textural features are extracted from these grey tone spatial dependence matrices which are used to identify the images.

2.7 Plant Species Identification Based on Neural Network

Authors: Lei Zhang, Jun Kong, Xiaoyun Zeng, Jiayue Ren
(Zhang et al., 2008)A plant identification method based on SOM (Self-organizing feature map) neural network is used because of its low learning complexity. Initially, the shape feature of the leaf is extracted by 2-D moment invariants and ratio of leaf's height to its width. Other than these features, the venation of the leaf is also captured as they are very useful in differentiating plant species. Secondly, the Discrete Wavelet Transform and statistical moments are employed to capture the texture information of venation. Finally, the SOM neural network is used to identify the species of the plant. Experimental results show that this method is feasible and effective in some plant species identification.

2.8 Deep Learning for plant identification using vein morphological Patterns

Authors: Guillermo L. Grinblat, Lucas C. Uzal, M´onica G. Larese and Pablo M. Granitto
(Grinblat et al., 2016)An application of deep learning in the area of agriculture, specifically plant identification based on leaf vein patterns. In this paper, they particularly classified between three legume species which are soybean, white bean and red bean. Instead of a task specific module for feature extraction, they have used a convolutional network. Because it is not necessary to handcraft a specific feature extraction method for this task. Using a simple visualization technique, they obtained the relevant vein patterns for the classification task performed by the deep model. The main result is that they obtained an improved accuracy using a standard deep learning model.

Chapter 3

ADDITIONAL INFO

3.1 Data Set

We used the data set available in https://data.mendeley.com/datasets/hb74ynkjcn/1 . We chose four crops namely Guava, Jamun, Lemon, Mango.

3.2 Software/Tools Requirements

The tech stack we used were:

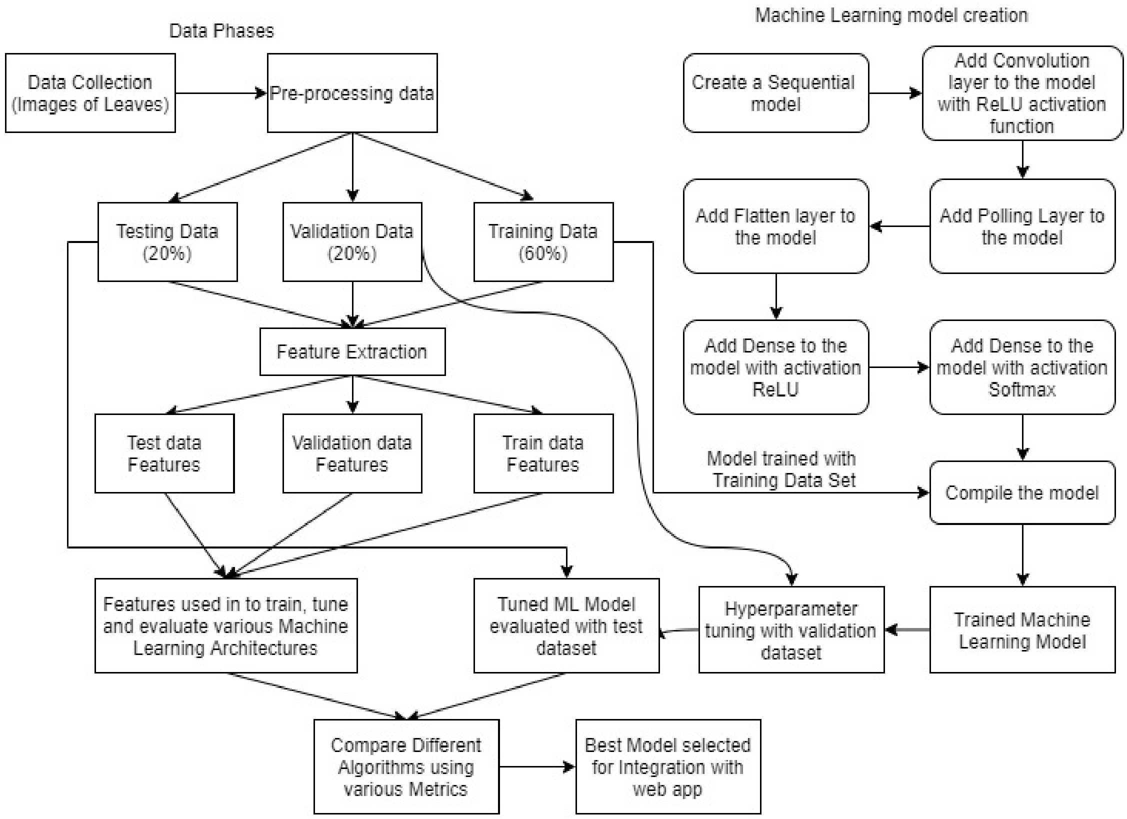
- HTML,CSS for frontend
- Django, Flask for backend
- Anaconda Juputer Notebook for IDE

Chapter 4

PROPOSED SYSTEM

4.1 System Analysis

The proposed architecture diagram is:



4.2 Description of modules

4.2.1 Data collection and data pre-processing

We gathered the dataset of plants like Jamun, Lemon, Mango, Guava from A Database of Leaf Images: Practice towards Plant Conservation with Plant Pathology.

We have implemented two pre-processing techniques namely, to remove the background using the cv2 module of python language, and to normalize the size of the

image data using PIL module.
We are splitting the available data set into test data which is 20% of the available data and the rest 80% will contribute to the train data for the CNN model we planned to implement.

4.2.2 CNN model

Creating a Sequential model

Sequential model is a Linear Stack of Layers where we create the model by stacking one layer upon another layer. This model is mostly suited in many deep learning problems.We use sequential model for our use-case since it is simple and easy to create a model and moreover, our model doesn't require any shared layers and requires only one layer at a time. To create a sequential model, the following steps are followed:

- Instantiate the model
- Stack layers using add()
- Compile the model

To add convolutional layer

This layer helps the filter learn and recognize features within the image. It recognizes edges, lines and other distinct shapes within the images. Hence convolutional step is also called Feature Mapping. We will have multiple channels and each channel will be trained to detect certain key features of the image.
The input image obtained is convolved using a filter (kernel) basically of size 3*3 or 5*5. The neural network learns the filter values of the filter that is used to detect the small and important features of the input, to match to the output. the filter (kernel) is glided through the input (receptive field) and element-wise product of the filter and input value is done and finally sum of those specific values is calculated for every sliding action to derive at the output.

To add pooling layer

The default pool size is (2, 2). We can also add our own pool size. This reduces the number of parameters in our network which is called as down-sampling and makes the model robust to detect features by making it impervious to orient and scale changes. Pooling layer involves down sampling of the obtained image. That is, the image is shrunk into smaller size. The network is further simplified by reducing the number of parameters. Various steps are involved in this layer. First, window size is selected (ideally 2 or 3). Second, stride is selected (usually 2). Third, iterate the window chosen across the entire image and take the maximum value.

To add flatten layer

The 2 dimensional array output after Pooling is flattened and converted into a 1 dimensional single long feature vector and passed as input to the fully connected layer where classification is undergone. Finally Sigmoid, Softmax are some of the activation functions applied to classify the output, in other words, identify the class of the object or image.

To add dense layer

Add the fully connected layer using Dense which is the last layer of CNN where neurons in this layer are fully connected with all the neurons of previous layers. Softmax is finally used as activation function for multi-class classification.

Compiling the model

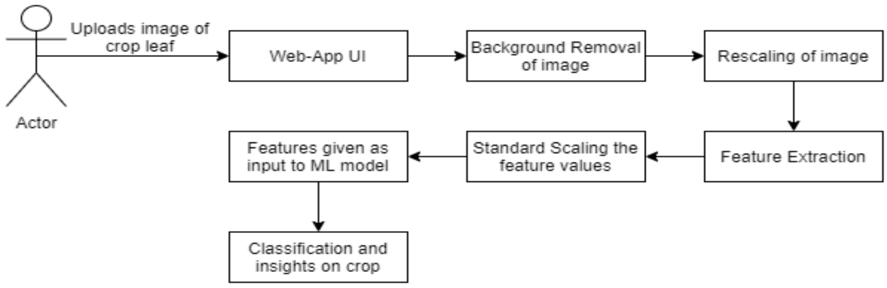
Compile defines the loss function, the optimizer and the metrics.
Loss function is used to find error or deviation in the learning process. We use "binary_crossentropy" for our loss function.
Optimization is an important process which optimize the input weights by comparing the prediction and the loss function. We use "Adam" optimizer for our CNN.
Metrics is used to evaluate the performance of your model. We find the reliability of the model using the '"accuracy" metric.

4.2.3 Training and testing the model

In this module, the split training data is used to train the compiled model. The model understands and trains itself with suitable weight values. The trained model is then fed with the testing data such that various metrics of the model can be evaluated.

Finally the UI for the web/mobile app will be created and the model created will be then integrated with the UI so that images can be captured and given such that the model identifies and displays the details of the plant.

4.2.4 End user interaction



4.2.5 Conclusion

We were able to create the CNN model with a 87.50% accurate prediction.

Chapter 5

RESULT AND DISCUSSION

The CNN model we created gave us a accuracy rate of 87.50%. Keeping this in mind, we wanted to try to solve the same problem using some other ML to increase the metrics and give a better prediction.

The metrics we consider were:

- Mean Square Error (MSE)
- Accuracy
- F1
- Precision
- Average run time

Model	MSE	Accuracy	F1	Precision	Avg. run time
XG Boosting	0.1225	0.9354	0.9357	0.9363	8.14s
Random forest	0.1886	0.7290	0.7241	0.7918	6.16s
Ada Boosting	0.6064	0.7612	0.7354	0.7817	5.45s
Bagging	0.1161	0.9419	0.9416	0.9421	5.61s
Naive Bayes	0.8967	0.6774	0.6733	0.6878	5.34s
SVM	1.0709	0.5483	0.5292	0.6339	5.44s
Logistic	0.8516	0.5741	0.5560	0.6556	5.47s

The best model of the lot is Bagging with a accuracy of 94.19%. So, we selected this as our final model and we performed hyper parameter tuning to boost the accuracy to a rate of 95.6% by using decision tree as base classifier, n_estimators being 101 and random states being 14.

Chapter 6

CONCLUSION

This paper studied various machine learning and deep learning approaches to learn discriminative features from leaf images with classifiers for plant identification. From the experimental results, we justified that learning the features through skimage and using a bagging model with a decision tree as a base classifier provides better results compared to other models. The observation of the performance of the model, when tested against a test dataset that had 4 different trees, was that it had an Accuracy of 95.6% and a mean squared error of 0.10, and the time taken to work in real-time is 5.61 sec. In future work, we will extend the work to recognize more species of trees.

Chapter 7

FUTURE ENHANCEMENT

We can further increase the scale of the project to identify leaves of many other plants and also try to create a model with much more accuracy and less MSE.

REFERENCES

1. Dave, S. and Runtz, K. "Image processing methods for identifying species of plants. IEEE.

2. Eid, H. F., Hassanien, A. E., and Kim, T.-H. (2015). "Leaf plant identification system based on hidden naïve bays classifier. IEEE.

3. Grinblat, G. L., Uzal, L. C., Larese, M. G., and Granitto, P. M. (2016). "Deep learning for plant identification using vein morphological patterns." 127, 418–424.

4. Lee, S. H., Chan, C. S., Wilkin, P., and Remagnino, P. (2015). "Deep-plant: Plant identification with convolutional neural networks. IEEE.

5. Varghese, B. K., Augustine, A., Babu, J. M., Sunny, D., and Cherian, S. (2020). "INFOPLANT: Plant recognition using convolutional neural networks. IEEE.

6. Wang, B., Brown, D., Gao, Y., and Salle, J. L. (2013). "Mobile plant leaf identification using smart-phones. IEEE.

7. Wang, X.-F., Huang, D.-S., Du, J.-X., Xu, H., and Heutte, L. (2008). "Classification of plant leaf images with complicated background." 205(2), 916–926.

8. Zhang, L., Kong, J., Zeng, X., and Ren, J. (2008). "Plant species identification based on neural network. IEEE.