International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072

A Plant Identification and Recommendation System

Vinoy Koshy Thomas[1], Jusbin Mathew[2], Nivin Emmanuel[3], Seban V Mathew[4]

[1,2,3,4]Student, Dept. of Computer Science and Engineering, Amal Jyothi College of Engineering, Kerala, India

---***---

Abstract - *Plants are the most essential part of the lives of people. As they make food from the carbon dioxide and releases oxygen into atmosphere. Oxygen is an essential need of humans. Plant recognition identifies the different plant and also recommends the type of plants that can be planted in that particular location. Recognition is done in three major steps: first, the preprocessing of the leaf image for noise reduction. Second, feature is extracted from the image. Third, identification of the plant from the given image using convolution neural network (CNN). This system is implemented as a Native application as well as a Web application.*

Key Words: **React-Native, Inception-v3, CNN, pH, Mobile Application,** etc

1. INTRODUCTION

Plants cover over 30 percent of earth land. A general definition of a plant is any organism that contains chlorophyll (a green pigment contained in a specialized cell called a chloroplast) and can manufacture its own food. The manufactured food is the oxygen which is the most essential part of living things. Automatic plant identification system is there for useful for people who are researching on plant and also for common people who don't know about a particular plant. Different properties which are used for identification is the shape, size, flower etc. Among all these factors leaf is the most important feature which can uniquely identify a plant.

Most of the plant recognition system is based on leaf identification. There are many algorithms used for leaf identification such as Multi-Layered Perceptron used to differentiate leaf boundaries and veins, Active Shape Model (ASM) to construct a leaf shape model that classifies query image and returns detection result. Main challenge in plant identification is: Images captured might be having variations like noise, brightness, color etc.

In this paper we are presenting a plant identification system:

1)Interface: An android app which is a user-friendly environment created for user interaction with the system.
2)Robustness: Variations in leaf images such as rotation and scaling should not affect the outcome of the result.
3)Portability: a website platform for user to interact with the system from any part of the world.

2. PROPOSED SYSTEM

The proposed system has mainly two functions: first, identification of a plant using the leaf image. Second, recommending the plants that grow in that particular location with respect to the identified plant. Identification can be summarized in to four different steps:

1) Import the color image of the leaf.
2) Image is preprocessed to remove noises.
3) Features of the images are extracted.
4) Comparing the features extracted with the existing database using CNN.

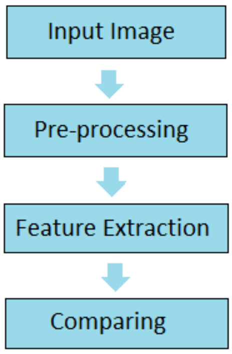

Fig -1: Overview of Recognition

2.1 Input Image

Image is inputted through the mobile application using the mobile camera or already taken image from mobile gallery [1]. Image will be automatically resized into 299*299-pixel size shown in Fig - 2. The color of leaf is usually green which should be converted into grey scale and also many other operations should be done over the image for identifying the image. This image is feed to pre-processing stage.

Fig – 2: Resizing of input image

2.2 Preprocessing

Image from the previous stage is an RGB image which is first converted into grey scale (Fig. 3). Then to reduce the 2-noise smoothing of an image is done using Gaussian filter. This filter is usually blurring the grey scale image using Gaussian function. The formula for Gaussian function in two dimensions is:

$$G(x, y) = \frac{1}{2\Pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

Image thresholding is done next which is the simplest method of image segmentation where binary image is created from the greyscale.[2][3] A morphological operation

Fig – 3: Greyscale conversion of input image

is done over binary image to creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image i.e. to close the holes in an image. Finally, the boundary of the image is extracted using contour.

2.3 Feature Extraction

In image processing, extraction feature is a type of reduction in dimensionality that effectively represents interesting parts of an image as a compact vector feature. This approach is useful when image sizes are large and in order to complete tasks such as image matching and retrieval, a reduced representation of features is required. Feature detection, extraction and matching are often combined to solve common computer vision problems such as object detection and recognition, image retrieval based on content, face detection and recognition, and texture classification.

The primary function of pattern recognition is to take an input pattern and assign it correctly as one of the possible output classes. The process can be subdivided into two: Feature selection and classification.[4] Extraction of features is an important step in building any classification of patterns and aims to extract the relevant information that characterizes each class. Relevant characteristics are extracted from objects to form feature vectors in this process. Classifiers then use these feature vectors to recognize the target output unit from the input unit. By looking at these features, it becomes easier for the classifier to classify between different classes as it makes distinguishing relatively easy.[5] Feature extraction is to find the set of parameters that accurately and uniquely define a character's shape. The main objective of extracting features is to extract a set of features that maximizes the recognition rate with the least number of elements and generates similar features for a variety of instances of the same symbol.

In our system, Contours of leaf are used to extract digital morphological features. The digital morphological feature includes basic geometric characteristics and morphological characteristics. Our system extracts five basic geometric characteristics i.e. the longest diameter, physiological length, physiological width, leaf area and perimeter of the leaf. Our system also extracts eleven digital morphological characteristics from geometric features as follows: smooth factor, aspect ratio, shape factor, narrow factor, diameter perimeter ratio, physiological length perimeter ratio and physiological width, and the remaining five are vein characteristics.[6][7]

2.4 Comparing

The leaves after feature extraction will be trained and based on the features a matrix is formed i.e. the tensor. The values in the tensor will be normalized to a value between 0 and 1. The leaf to be recognized undergoes the same processing as the training data. It will also be converted into tensor and the values are to be normalized first. The new tensor is compared with already trained leaves one by one.

For optimizing, we use Stochastic Gradient Descent (SGD). The word `stochastic' means a random probability-related system or process. Therefore, a few samples are selected randomly in Stochastic Gradient Descent instead of the entire data set for each iteration. The standard gradient descent algorithm updates the parameters of the objective J() in Gradient Descent as,

$$\theta = \theta - \alpha \bigtriangledown_\theta \theta E[J(\theta)] \quad (2)$$

where expectations are approximated in the above equation by evaluating the cost and gradient over the complete training set. Stochastic Gradient Descent (SGD) simply eliminates the expectations in the update and calculates the parameter gradient using just one or a few examples of training. The new update will be provided by,

$$\theta = \theta - \alpha \bigtriangledown_\theta (\theta; x^{(i)}, y^{(i)}) \quad (3)$$

with a pair

$$(\theta; x^{(i)}, y^{(i)})$$

from the training set. The learning rate α in batch gradient descent is typically much lower than a corresponding learning rate because there is much greater variance in the update. It can be quite difficult to choose the appropriate learning rate and schedule (i.e. to change the value of the learning rate as learning progresses). A standard method that works well in practice is to use a small sufficiently constant learning rate that gives stable convergence in the initial epoch (full pass through the training set) or two of training and then halve the learning rate value as convergence slows down. An even better approach is to evaluate a set out after each epoch and anneal the learning rate when a small threshold is below the change in objective between epochs. This tends to give a good local optimum convergence. Another common schedule is to anneal the learning rate as ab+t at each iteration where a and b dictate the initial learning rate and when the annealing begins respectively.

In this paper, we use inception model for training images. Inception - v3 network model is a deep neural network, it's very hard for us to train it directly with a low configuration computer, it takes at least a few days to train it. So, we used Google Collab as the platform for the convolutional neural network.[8][9] We use the transfer learning method that keeps the previous layer parameters and removes the last layer of the Inception - v3 model, then retrains the last layer. In the last layer, the number of output nodes is equal to the number of categories in the dataset. The dataset which we use is the leaf snap dataset containing 184 classes. But the dataset contains irrelevant plants according to our location so we only chose 10 plants that are in our location. So, in our Inception - v3 model, the last layer has 10 output nodes.

3. RECOMMENDATION SYSTEM

Our recommendation system generates one or more recommendations typically for the user. The system will make automatic input-based predictions using collaborative filtering method.[10] There are corresponding pH, temperature and rainfall values (Fig 4) for each plant in the dataset, a new range table is created on the basis of these values. The data is represented as a range of values in the new table (e.g. pH 4-7) along with the names of the plants that fall within the specifies range.

Once a plant is identified, its pH, temperature, and rainfall values are obtained. The system then identifies the ideal range within the range table to which the plant belongs. Once the ideal range has been obtained, the user will be shown the list of other plants falling within the same range in the range table. Our system would say the displayed plants are more likely to grow in the location of the user.

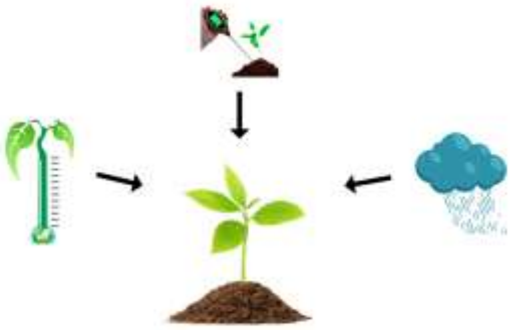

Fig – 4: Factors defining recommendation

4. Experimental Analysis

We created an application to facilitate user interaction and integrate the entire system into the application. The process begins with the user taking an input image via the application either using device camera or manually selecting image from the gallery. The system recognizes the plant after selecting the picture. Similar plants are identified from the database by analyzing factors such as pH, temperature and rainfall of the recognized plant and both are shown on the screen.

In this experiment 10 leaf images are given, out of which 7 plants are correctly classified together with the recommended plants. In addition, more plant images were given as inputs and the system was approximately 70% accurate.

5. CONCLUSIONS

In this paper, we demonstrated a plant recognition and recommendation system using the Convolutional Neural Network (CNN) inception-v3 model. The systems available are based on prediction of recognition and disease. The system demonstrated has a recommendation feature that can be used to identify similar plants that can be planted in that area.

We think further improvements can be done in the system:

1) A separate binary classifier can be included for identifying leaf images from other images
2) More number of local plants can be included in the database
3) For improving accuracy more data can be used for training and testing.

REFERENCES

1. Wang, D. Brown, Y. Gao and J. L. Salle, "Mobile plant leaf identification using smart-phones," IEEE ICIP 2013

2. J. S. Weszka," A Survey of Threshold Selection Techniques", Computer Vision Graphics and Image Processing, vol. 7, pp. 259-265, 1978.

3. Krishna Kant Singh, Akansha Singh," A Study of Image Segmentation Algorithms for Different Types of Images", IJCSI International Journal of Computer Science Issues 2010, 2010.

4. J. Kittler. Feature set search algorithms. In C. H. Chen, editor, Pattern Recognition and Signal Processing, pages 41-60. Sijthoff and Noordhoff, Alphen aan den Rijn, Netherlands, 1978.

5. Yao, J.T. and Yao, Y.Y., Information granulation for Web based information retrieval support systems, Data Mining and Knowledge Discovery: Theory, Tools, and Technology V, Orlando, Florida, USA, April 21- 22, 2003, Dasarathy, B.V. (Ed.), The International Society for Optical Engineering, pp. 138-146, 2003.

6. Bhardwaj, M. Kaur, "A review on plant recognition and classification techniques using leaf images," International Journal of Engineering Trends and Technology, Volume 4, Issue 2, 2013.

7. Shabanzade, M.M.Zahedi and S.A.A. ghvami, "Combination of local descriptors and global features for leaf recognition" Signal and Image processing: International Journal, vol.2: 23-21. DOI: 10. 5121/SIPIJ.

8. Jassmann TJ, Tashakkori R, Parry RM (2015) Leaf classification utilizing a convolutional neural network. In: SoutheastCon.

9. A Krizhevsky, I Sutskever, G E. Hinton," ImageNet Classification with Deep Convolutional Neural Networks[J]", Advances in Neural Information Processing Systems 2012, vol. 25, no. 2, 2012.

10. Alexandrin Popescu and Lyle H. Ungar, David M. Pennock and Steve Lawrence," Probabilistic Models for Unified-Collaborative and Content Based Recommendation in Sparse-Data Environments", POPESCUL ET Al,2001.