PLANT SEEDLINGS CLASSIFICATION USING IMAGE PROCESSING AND DEEP CONVOLUTIONAL NEURAL NETWORK

1st Chinmay Khedkar

UG student,Computer
Pune,India
crk.datascience@gmail.com

4th Saurabh Manoj Gore
Pune,India
saurabhgore28@gmail.com

7 th Dhruv Mayenkumar Bhavsar
Petlad,India
dhruvbhavsar225@gmail.com

10 th Kshitij Gupta
Pune,India
kshitijgupta139@gmail.com

2nd Aishwarya Jagannath Bargale
Solapur, India
aishwaryabargale@gmail.com

5th Utkarsha Sanjay Kandale
Solapur,India
uskpra2223@gmail.com

8 th Dilip Bhagavan Sonavane
Pune, India
dipusonavane@gmail.com

11 th Harsh Mahajan
Pune,India
harsh321mahajan@gmail.com

3rd Deepali Hanamant Sankappanavar
Solapur,India
deepalisankappanavar2399@gmail.com

6th Avani Manish Sanghvi
Vadodara,India
amsanghvi123@gmail.com

9 th Deborshi Deb
Shillong, Meghalaya
debdeborshi123@gmail.com

***Abstract*— In order to improve the farming process, there is need to classify the plant seedlings. The target of this project is to distinguish between the different weed seeding and crop seeding of 12 different plant species, hence, it's a multi-classification problem. As we take a image of a crop seedling and output the correspondent class from our 12 classes. We'll be using CNN model to classify the plants.**

Keywords—Image Processing, Plant seedlings, plant species recognization, precision agriculture, machine vision,machine learning.

INTRODUCTION

The Plants exist everywhere we live, as well as places without us. Many of them carry significant information for the development of human society. The urgent situation is that many plants are at the risk of extinction. So it is very necessary to set up a database for plant protection. We believe that the first step is to teach a computer how to classify plants.

Sampling leaves and photoing them are low-cost and convenient. One can easily transfer the plant seedling image to a computer and a computer can extract features automatically in image processing techniques. But it is not easy to extract and transfer those features to a computer automatically. This paper tries to prevent human interference in feature extraction.

According, to data acquisition from living plant automatically by the computer has not been implemented. This paper implements a leaf recognition algorithm using easy-to-extract features and high efficient recognition algorithm. Our main improvements are on feature extraction and the classifier. All features are extracted from digital leaf image. As to the classifier, we use PNN for its fast speed and simple structure. The whole algorithm is easy to-implement, using common approaches.

EASE OF USE

The ultimate goal is to build applications that could use machine learning and vision.
for process automation that could increase the productivity and ease the burden of the every- day life.
There are multiple tasks at hand that could use image processing and machine vision. Some of them are: Object recognition in images , image segmentation to extract regions of interest, image database search etc.

All of these tasks are equally important for the development of the computer science and intelligent systems where these approaches could be beneficial.

METHODOLOGY

Implementation:

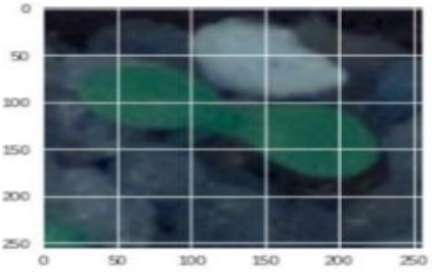
The implementation process can be split into two main stages:

- Preprocessing data stage
- The classifier training stage

Preprocessing:

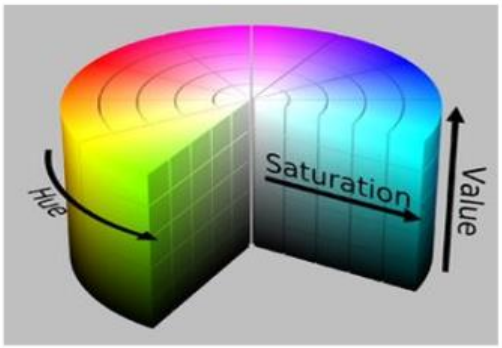
1- Resizing Images:

Images have not the same size images have been resized the images to 256*256 pixel to feed it later to the neural network In this figure the image after resize.



2- Creating mask for the images :

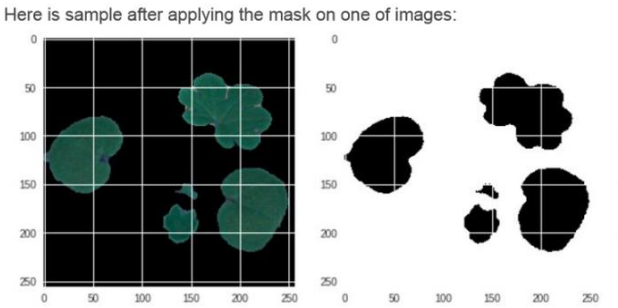
create_mask_for_plant function returns an image mask: Matrix with shape (image_height, image_width). In this matrix there are only 0 and 1 values. The 1 values define the interesting part of the original image. I can create this mask using HSV of the image.
The HSV color-space is suitable for color detection because with the Hue we can define the color and the saturation and value will define "different kinds" of the color. (For example it will detect the red, darker red, lighter red too). We cannot do this with the original BGR color space
This figure illustrate HSV space of the image.



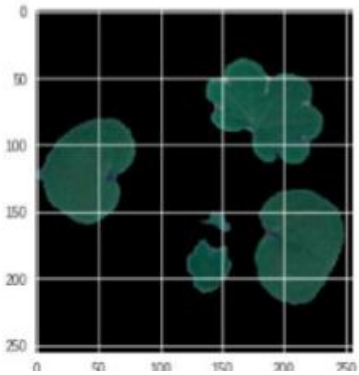
After converting RGB to HSV I started to apply morphological operations

3-morphological operations :
one of most common morphological operation is closing : closing used to close the small halls in the images. This figure below illustrate the image before and after applying closing:

Here is sample after applying the mask on one of images:

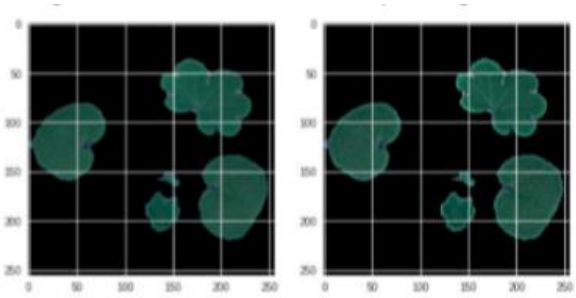


4- Segmentation: Segmentation is partitioning an image into distinct regions containing each pixels with similar attributes. To be meaningful and useful for image analysis and interpretation, the regions should strongly relate to depicted objects or features of interest. Here is a sample after applying segmentation on one of images:



5- Sharpening:

Sharpening an image increases the contrast between bright and dark regions to bring out features. Image before and after sharpening:



6- Splitting data into training and testing set:

In this step I used sklearn train_test split 70% for training and 30% for testing Then data is split into testing and validation set 50% for testing and 50% for validation. The point of using the validation set is to try to avoid overfitting and check if an overfitting may occur.

The classifier training stage:

During the first stage, the classifier was trained on the preprocessed training data. This was done in a Jupyter notebook (titled "FINAL_seedling.ipynp"), and can be further divided into the following steps:

1. Load both the training and validation images into memory, preprocessing them as described in the previous section the data required for a single training step (a batch of images, their labels, and the learning rate)
2. Define the network architecture and training parameters
3. Define the loss function, accuracy
4. Train the network, logging the validation/training loss and the validation accuracy
5. Plot the logged values
6. If the accuracy is not high enough, return to step 3

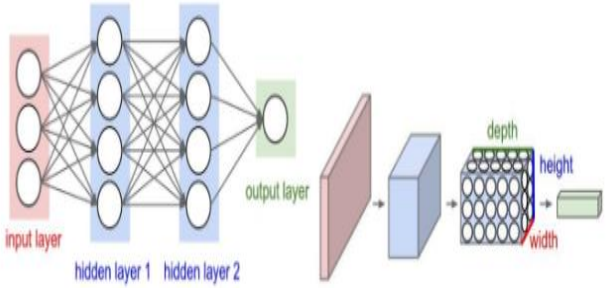
7. Save and freeze the trained network

Algorithm And Technique

CNN Architecture:

In general, Neural networks accept a single vector as input, transform it to a series of hidden layers, which in turn is made up of set of neurons that are fully connected to all neurons in the previous layer. Neurons of the same layer are independent and do not share any connections. After the hidden layers, is the last fully connected layer which is also called the 'output layer', where each node outputs score for each class. The downside of regular neural network is that they don't scale well to full images. It's mainly because with images of decent size, the number of neurons and weights that the network must accommodate becomes unmanageable. This is where Convolutional Neural Network comes to rescue with its neurons arranged in 3 dimensions (width, height, depth). Each of the layer in CNN accepts 3D input volume and transforms it into 3D output volume.

Following is a simple visualization of how CNN arranges its neurons in 3 dimensions (width, height, depth):



Following are the layers that are used to build a CNN: Input layer (w,h,d) Input layer of shape (w,h,d) represents image of size 'w x h' and 'd' number of color channels. For example, for an image of size (256x256) with 3 color channels (HSV), the input layer will hold raw pixel values of the image as a vector of size [256,256,3].

CONV layer:

The CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. For eg, if we use 32 filters, CONV layer will output a volume equal to (256x256x3) Activation functions Activation layer will apply an element wise activation functions, leaving the volume unchanged.

POOL layer:

Pool layer performs downsampling operation along the spatial dimensions (width, height), outputting a reduced volume than the previous layer. For eg, (128x128x32).

Fully-Connected Layer:

FC layer, also called as the dense layer, each neuron will be connected to all the neurons of the previous layer. FC layer when used as the output layer results in much reduced volume of size [1x1xm], where m is the number of categories that are to be predicted. Each of the nodes of fully connected layer outputs a score corresponding to a class score.

Dropout layer

Dropout layer is used as a method of regularization to combat over-fitting of the training set. It 'drops' neurons at random (depending on the probability mentioned) while calculating the forward prop and backward prop, resulting in a simpler version of the CNN for each iteration and hence giving the model a hard time to overfit the training set. Hence for

CNN algorithm

consists of several convolution (CNV) operations followed of the image sequentially which is followed by pooling operation (PL) to generate the neurons feed into fully connected (FC) layer.

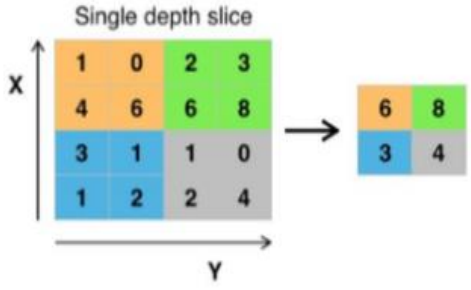
We used the Keras Sequential API, where we have just to add one layer at a time, starting from the input.

Input: of CNV is typically 2D image data with HSV (hue saturation value) .

The first is the convolutional (Conv2D) layer: It is like a set of learnable filters. We chose to set 32 filters for the two firsts conv2D layers and 64 filters for the 2nd convolutional layer and 128 filters for the two last ones. Each filter transforms a part of the image (defined by the kernel size) using the kernel filter. The kernel filter matrix is applied on the whole image. Filters can be seen as a transformation of the image. The CNN can isolate features that are useful everywhere from these transformed images (feature maps).

The second important layer in CNN is the pooling (MaxPool2D) layer. This layer simply acts as a down sampling filter. It looks at the 2 neigh-boring pixels and picks the maximal value. These are used to reduce computational cost, and to some extent also reduce over-fitting. We have to choose the pooling size (i.e the area size pooled each time) more the pooling dimension is high, more the down sampling is important.

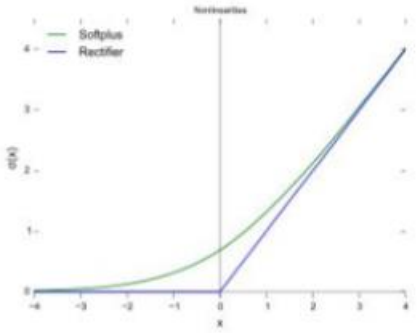
This figure below illustrate Max pooling with a 2x2 filter and stride = 2:



Combining convolutional and pooling layers, CNN are able to combine local features and learn more global features of the image. Dropout is a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their weights to zero) for each training sample. This drops randomly a proportion of the network and forces the network to learn features in a distributed way. This technique also improves generalization and reduces the overfitting.

Relu Layer :

Relu is the abbreviation of Rectified Linear Units. This layer applies the non-saturating activation function. (x) x max(0,) f = + = x It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. Figure below illustrate Relu function: Plot of the rectifier (blue) and softplus (green) functions near x = 0



The Flatten layer is use to convert the final feature maps into a one single 1D vector. This flattening step is needed so that you can make use of fully connected layers after some convolutional/maxpool layers. It combines all the found local features of the previous convolutional layers. In the end i used the features in two fully-connected (Dense) layers which is just artificial neural networks (ANN) classifier. In the last layer (Dense(10,activation="softmax")) the net outputs distribution of probability of each class.

Fully Connected (FC) Layer: This layer will reduce the size of input data to the size of classes that the CNN is trained for by combining output of CNV layer with different weights. Each neuron at the output of the CNV layer will be connected to all other neurons

after weighted properly, Similar to CNV layer, weight of these taps in FC layer is found though back propagation algorithm.

Classification Layer (CL): This is the final layer of the CNN that converts the output of FC to probability of each object being in a certain class. Typically soft-max type of algorithms are used in this layer.

Refinement:

- Decreasing learning rate from .01 to .001 avoid the over-fitting of the model.
- By adding data augmentation accuracy increased to : 91%
- Adding some additional layer also improved the model
- Changing dropout from .25 to .3 improved the accuracy for 86%

Future Scope

As an improvement and future work, We would like to try data masking on the training set. Noise from the background of the images can be cancelled by masking images. We believe that without the background noise and restricting the visibility to the green leaves, the model can be trained better, and we may notice significant improvement in the performance.
Dataset collection may involve diseases of different season and on different land. The proposed system can be further improved by incorporating other high level features such as shape features (Zernike moments, higher order statistics, counterlet transforms etc).

Application And Advantages

Applications:

1.As biology includes number of plant species so it become difficult to identify them so by using this model we ca n easily differentiate among several species of plant seedlings.

2.As we seen in future scope we will able to identify diseases,characteristics of seedlings and other necessary information related to it. With help of this we can increase rate of production.

Advantages:

1.Convinient way to classification and less time consuming for classification as compared to manual classification.

2.Better Algorithms are designed to give the high accuracy.

Conclusion

This paper introduces a neural network approach for plant seedlings recognition. The computer can automatically classify different kinds of plants via the leaf images loaded from digital cameras or scanners.

PNN is adopted for it has fast speed on training and simple structure. 12 features are extracted and processed by PCA to form the input vector of PNN. Experimental result indicates that our algorithm is workable with an accuracy greater than 90% on different kinds of plants. Compared with other methods, this algorithm is fast in execution, efficient in recognition and easy in implementation. Future work is under consideration to improve it.

References

1. http://scholar.google.co.in/scholar?q=IEEE+paper+for+classification+of+plants+seedlings+using+machine+learning&hl=en&as_sdt=0&as_vis=1&oi=scholart
2. J.-X. Du, X.-F. Wang, and G.-J. Zhang, "Leaf shape based plant species recognition," Applied Mathematics and Computation, vol. 185, 2007.
3. Y. Ye, C. Chen, C.-T. Li, H. Fu, and Z. Chi, "A computerized plant species recognition system," in Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, October 2004.
4. D. Warren, "Automated leaf shape description for variety testing in chrysanthemums," in Proceedings of IEE 6th International Conference Image Processing and Its Applications, 1997.
5. T. Brendel, J. Schwanke, P. Jensch, and R. Megnet, "Knowledge-based object recognition for different

morphological classes of plants," Proceedings of SPIE, vol. 2345, 1995
6. D. F. Specht, “Probabilistic neural networks,” Neural Networks, vol. 3, 1990.
7. M. T. Hagan, H. B. Demut, and M. H. Beale, Neural Network Design, 2002.
8. (2007) Matlab neural network toolbox documentation. MathWorks. Inc. [Online]. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/radial10.html#8378
9. D. F. Specht, “Probabilistic neural networks for classification mapping, or associative memory,” in Proceedings of IEEE International Conference on Neural Networks, vol. 1, 1988.
10. Motoyoshi, S. Nishida, L. Sharan, and E. H. Adelson, “Image statistics and the perception of surface qualities,” Nature, vol 447, May 2007
11. M. J. Dallwitz, “A general system for coding taxonomic descriptions,” Taxon, vol. 29, 1980.
12. H. Fu, Z. Chi, D. Feng, and J. Song, “Machine learning techniques for ontology-based leaf classification,” in IEEE 2004 8th International Conference on Control, Automation, Robotics and Vision, Kunming, China, 2004
13. D. Warren, “Automated leaf shape description for variety testing in chrysanthemums,” in Proceedings of IEE 6th International Conference Image Processing and Its Applications, 1997.
14. T. Brendel, J. Schwanke, P. Jensch, and R. Megnet, “Knowledgebased object recognition for different morphological classes of plants,” Proceedings of SPIE, vol. 2345, 1995.