PUNE INSTITUTE OF COMPUTER TECHNOLOGY,

DHANKAWADI PUNE-43.

A

Seminar Report

On

PLANT DISEASE CLASSIFICATION USING NEURAL NETWORKS

SUBMITTED BY

Madhura Pawar

ROLL NO:3256

CLASS:TE 2

GUIDED BY

Prof. R.R.Patil

ISO 9001: 2015 Certified

COMPUTER ENGINEERING DEPARTMENT

Academic Year:2017-18

P:F-SMR-UG/08/R0

Table of Contents

Sr. No	Topic	Page No.
1	Abstract	4
2	Introduction	5
3	Related Work	5
4	Literature Survey	6
5	Survey of Mathematical Models	7
6	Proposed Mathematical Model	8
7	Design and Analysis of System	10
8	Discussion on Implementation Results	12
9	Conclusion and Future Enhancement	13
10	References	14

Plant Disease Classification Using Neural Networks

Abstract:
Indian economy mostly depends on agriculture. Hence increasing quality production has become necessary day by day.Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product.It is very difficult to monitor the plant diseases manually.Hence,image processing is used for the detection of plant diseases. Disease detection involves the steps like image acquisition, image pre-processing, image segmentation, feature extraction and classification.The classification is being done through a convolutional neural network classifier.

Keywords:
Neural Networks,Image Classification,Image Processing,Machine Learning,Plant Diseases.

INTRODUCTION

India is an agricultural country. Farmers have large range of diversity for selecting various suitable crops and finding the suitable pesticides for plant. Disease on the plant leads to the significant reduction in both the quality and quantity of agricultural products. Monitoring of health and disease on plant plays an important role in successful cultivation of crops in the farm. In early days, the monitoring and analysis of plant diseases were done manually by the expertise person in that field. This required a tremendous amount of work and excessive processing time. In most of the cases disease symptoms are seen on the leaves, stem and fruit. The plant leaf for the detection of disease is considered which shows the disease symptoms.

Various efforts have been developed to prevent crop loss due to diseases. Historically, disease identification has been supported by agricultural extension organizations or other institutions such as local plant clinics. In more recent times, such efforts have additionally been supported by providing information for disease diagnosis online, leveraging the increasing internet penetration worldwide. Recently, tools based on mobile phones have increased in large amounts, taking advantage of the fast uptake of mobile phone technology in all parts of the world.

RELATED WORK

In [8], authors present, review, and recognize the demand for developing a rapid, cost-effective, and reliable health-monitoring sensor that facilitates advancements in agriculture. They described the currently used technologies that include spectroscopic and imaging-based and volatile profiling-based plant disease detection methods for the purpose of developing ground-based sensor system to assist in monitoring health and diseases in plants under field conditions. Numerous procedures are currently in use for plant disease detection applying computer vision. One of them is disease detection by extracting colour feature as authors in [9] have presented. In this paper YcbCr, HSI, and CIELB colour models were used in the study; as a result, disease spots were successfully detected and remained unaffected by the noise from different sources, such as camera flash.

There are some approaches which apply the feed-forward back propagation of neural networks consisting of one input, one output, and one hidden layer for the needs of identifying the species of leaf, pest, or disease; this model was proposed by the authors in [10]. They developed a software model, to suggest remedial measures for pest or disease management in agricultural crops.

LITERATURE SURVEY

The following table shows the literature survey by comparing techniques proposed in various references:

Sr No.	Author and year	Paper Name	Methodology used	Limitations	Adv.
1	Emanuel Cortes,2017[1]	'Plant Disease Classification Using Convolutional Networks and Generative Adversarial Networks'	Classification model was implemented using both CNN as well as GANs.	Accuracy decreases when environmental conditions of the leaf images are changed.	Use of GANs holds a promise to increase the accuracy of the model.
2	Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic,2016[2]	'Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification'	CNN model with 5 convolutional and 3 fully connected layers in CaffeNet were used.	Disease classification is done only on basis of upper section of the leaves.	Overall accuracy of 96.3% was achieved.
3	Mohanty Sharada P., Hughes David P., Salathé Marcel,2016[3]	'Using Deep Learning for Image-Based Plant Disease Detection'	AlexNet and GoogLeNet architectures were used.	Accuracy reduced when tested on conditions different from images used for training.	Holds the future scope of using this technique in smart phones.

4	Vijai Singh, A.K. Misra,2017[5]	'Detection of plant leaf diseases using image segmentation and soft computing techniques'	Segmentation using genetic algorithm and SVM classifier used.	Recognition rate needs to be improved using ANN, Bayes classifier,etc .	With very less computational efforts the optimum results were obtained.

SURVEY OF MATHEMATICAL MODELS

1. Generative Adversarial Networks:

GAN consists of two models:

- A discriminator *D* estimates the probability of a given sample coming from the real dataset. It works as a critic and is optimized to tell the fake samples from the real ones.
- A generator *G* outputs synthetic samples given a noise variable input *z* (*z* brings in potential output diversity). It is trained to capture the real data distribution so that its generative samples can be as real as possible, or in other words, can trick the discriminator to offer a high probability.

Given,

Symbol	Meaning	Notes
Pz	Data distribution over noise input *z*	Usually, just uniform.
pg	The generator's distribution over data *x*	
pr	Data distribution over real sample *x*	

On one hand, we want to make sure the discriminator *D*'s decisions over real data are accurate by maximizing $E_{x \sim pr(x)}[\log D(x)]$. Meanwhile, given a fake sample $G(z), z \sim pz(z)$, the discriminator is expected to output a probability, $D(G(z))$, close to zero by maximizing $E_{z \sim pz(z)}[\log(1-D(G(z)))]$.

P:F-SMR-UG/08/R0

On the other hand, the generator is trained to increase the chances of D producing a high probability for a fake example, thus to minimize $E_{z \sim p_z(z)}[\log(1-D(G(z)))]$

When combining both aspects together, D and G are playing a minimax game in which we should optimize the following loss function:

$\min_G \max_D L(D,G) = E_{x \sim p_r(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1-D(G(z)))] = E_{x \sim p_r(x)}[\log D(x)] + E_{x \sim p_g(x)}[\log(1-D(x)]$

2. Convolutional Neural Networks:

Suppose that we have some $N \times N$ square neuron layer which is followed by our convolutional layer. If we use an $m \times m$ filter ω, our convolutional layer output will be of size $(N-m+1) \times (N-m+1)$. In order to compute the pre-nonlinearity input to some unit x^{ℓ}_{ij} in our layer, we need to sum up the contributions (weighted by the filter components) from the previous layer cells:

$$x^{\ell}_{ij} = \omega_{ab} y^{\ell-1}_{(i+a)(j+b)}$$

Then, the convolutional layer applies its nonlinearity:

$y^{\ell}_{ij} = \sigma(x^{\ell}_{ij})$.

Let's assume that we have some error function, E, and we know the error values at our convolutional layer. Note that the error we know and that we need to compute for the previous layer is the partial of E with respect to each neuron output $(\partial E/\partial y^{\ell}_{ij})$.

Let's first figure out what the gradient component is for each weight by applying the chain rule. Note that in the chain rule, we must sum the contributions of *all* expressions in which the variable occurs.

3. SVM Classification:

Support vector machines are a supervised learning method used to perform binary classification on data.

For **linear kernel** the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

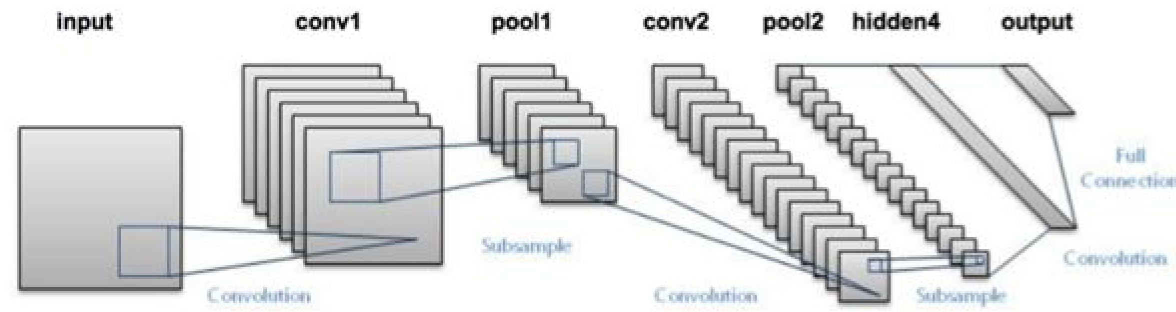
f(x) = B(0) + sum(ai * (x,xi))

The **polynomial kernel** can be written as: *K(x, xi) = 1 + sum(x * xi)^d*

The exponential kernel is written as: *K(x,xi) = exp(-gamma * sum((x — xi²)))*

PROPOSED MATHEMATICAL MODEL

The proposed model of the convolutional neural network is put before:



We first accept a volume $W1 \times H1 \times D1$. 4 hyper parameters are required:

1. Number of filters k
2. Their spatial extent F
3. The stride S
4. The amount of zero padding P.

Produces a volume of size $W2 \times H2 \times D2$ where:

- $W2=(W1-F+2P)/S+1$
- $H2= (H1-F+2P)/S+1$
- $D2=K$
- With parameter sharing, it introduces $F \cdot F \cdot D1$ weights per filter, for a total of $(F \cdot F \cdot D1)$ K weights and K biases.
- In the output volume, the d-th depth slice (of size $W2 \times H2$) is the result of performing a valid convolution of the d-th filter over the input volume with a stride of S, and then offset by d-th bias.

Back propagation can be separated into 4 distinct sections:

1. Forward pass
2. Loss function: There are various loss functions but the most common used one is MSE (mean squared error):

$$E_{total} = \Sigma \text{(target-output)}^2/2$$

3. Backward pass: Here the weights which contributed most to the loss are found and ways to adjust them so that the loss decreases are evaluated.
4. Weight update: Finally the weights are updated, so that the global minimum is reached.

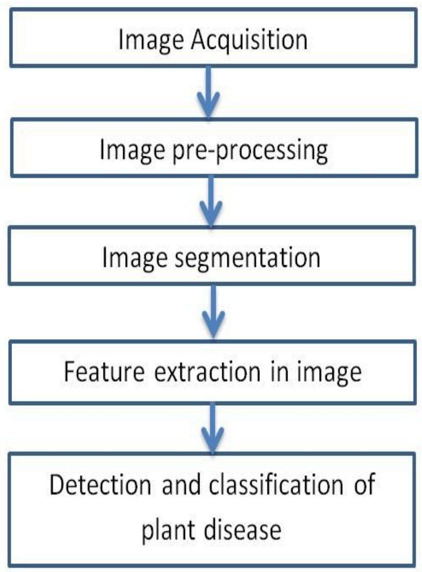
P:F-SMR-UG/08/R0

$$W = w_i - \alpha \; dL/dw$$

Where α is the learning rate.

DESIGN AND ANALYSIS OF SYSTEM

The basic steps for plant disease detection and classification are as follows:



1.Image Acquisition:

The dataset was obtained from [11]. Analysis of 1732 images of plant leaves which have a spread of 4 class labels is done.The 4 class labels are as follows:Apple Black Rot,Apple Healthy,Corn Common Rust,Corn healthy.

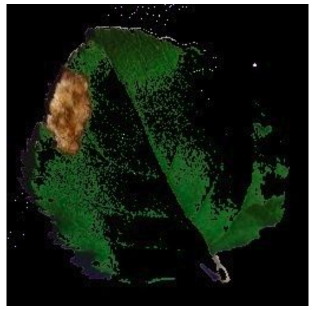
2.Image pre-processing:

The images were resized to 256*256 pixels using OpenCV framework.A python script was written to resize the images.

3.Image segmentation:

Segmentation means partitioning of image into various part of same features or having same similarity.The RGB image is first converted to HSV image.The background pixels and the green pixels are masked out using thresholding techniques to obtain a segmented image.The segmentation was done in OpenCV framework.

Before Segmenting



After segmenting

4.Data Augmentation:

The main purpose of applying augmentation is to increase the dataset and introduce slight distortion to the images which helps in reducing overfitting during the training stage.

Variations introduced:

- Rotation range:30□°
- Horizontal shifting range:0.1
- Vertical shifting range:0.1
- Random flip:Horizontal

5.Feature extraction in image(Training neural network):

- Specifications of convolutional neural network:
  - 5.1 Convolutional Layer (Filters:20,Kernel Size:5*5)
  - 5.2 Activation Layer(ReLu)
  - 5.3 MaxPooling(Pool Size:2*2,Strides:2*2)
  - 5.4 Convolutional Layer (Filters:50,Kernel Size:5*5)
  - 5.5 Activation Layer(ReLu)
  - 5.6 MaxPooling(Pool Size:2*2,Strides:2*2)
  - 5.7 Fully connected layer with 500 units
  - 5.8 Activation Layer(ReLu)
  - 5.9 Fully connected layer with specified no. of classes
  - 5.10 Activation Layer(SoftMax)
- Epochs:25
- Learning rate:1e-3
- Test-train ratio:1:3
- Optimiser:Adam Optimiser

6.Detection and Classification of Plant Diseases

The input for testing is an image and the output obtained is a label which tells to which class it belongs and with what accuracy.

Proposed Algorithm for Training:

1.Start
2. For every image in the image_dataset
   2.1 Start
   2.2. Resize the image to 28*28 pixels
   2.3. Convert the pixels into image intensity values.
   2.4 Normalize the integer array obtained.
   2.5 label := image_path_name
   2.6 if label contains “class1”
         2.6.1 label_ := 0;
       else if label contains “class2”
         2.6.2 label_ := 1;
       else if label contains “class3”
         2.6.3 label_ := 2;
       else
         2.6.4 label_ := 3;
         2.6.5 End
3. Split the dataset in an appropriate train-test ratio.
4. Perform image augmentation.
//Initialize the model
5. model := Lenet.build()
   model.compile()
//Train the network
6. model.fit_generator()
//Save the model
7.model.save(file_name)
8.End

DISCUSSION ON IMPLEMENTATION RESULTS

The convolutional neural network model was trained on a model with 2 convolutional layers with 32 filters having 5*5 kernel size,2 pooling layers,3 relu activation layers,1 softmax layer and a dense layer. The model was trained till about 50 epochs and the following Loss/Accuracy vs Epochs graph was obtained:

The following graph indicates that with increasing no. of epochs the model becomes more fine tuned as the loss decreases.Hence,the overall accuracy obtained was **97.35%.**

```
Epoch 45/50
42/42 [==============================] - 4s 99ms/step - loss: 0.0590 - acc: 0.9784 - val_loss: 0.0958 - val_acc: 0.9669
Epoch 46/50
42/42 [==============================] - 4s 98ms/step - loss: 0.0527 - acc: 0.9829 - val_loss: 0.0558 - val_acc: 0.9779
Epoch 47/50
42/42 [==============================] - 5s 113ms/step - loss: 0.0627 - acc: 0.9791 - val_loss: 0.3004 - val_acc: 0.9161
Epoch 48/50
42/42 [==============================] - 5s 112ms/step - loss: 0.0532 - acc: 0.9844 - val_loss: 0.0665 - val_acc: 0.9669
Epoch 49/50
42/42 [==============================] - 5s 127ms/step - loss: 0.0478 - acc: 0.9844 - val_loss: 0.1099 - val_acc: 0.9558
Epoch 50/50
42/42 [==============================] - 5s 118ms/step - loss: 0.0750 - acc: 0.9769 - val_loss: 0.0571 - val_acc: 0.9735
```

Screenshot of model getting trained displaying the training and validation losses and accuracies for last 4 epochs.

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

There are a number of limitations at the current stage that need to be addressed in future work.First,when tested on a set of images different from the images used for training ,the model's accuracy is reduced substantially.The second limitation is that we are currently constrained to the classification of single leaves,facing up,on a homogenous background.While these are straight forward conditions,a real world application should be able to classify images of a disease as it presents itself directly on the plant.Thus,new image collection efforts should try to obtain images from different perspectives and ideally from settings that are as realistic as possible.

Future enhancement:

The approach of training deep learning models on increasingly large and publicly available image datasets presents a clear path towards smartphone-assisted crop disease diagnosis on a massive global scale.Smartphones in particular offer very novel approaches to help identify diseases because of their tremendous computing power,high-resolution displays and extensive built-in sets of accessories such as advanced HD cameras.While the training of the model takes a lot of time,the classification is very fast(less than a second on a CPU) and can thus be easily be implemented on a smartphone.

REFERENCES

[1] A report on ‘Plant Disease Classification Using Convolutional Networks and Generative Adversarial Networks’ by Emanuel Cortes. July 2017, Stanford University.

[2] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic, “Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification,” Computational Intelligence and Neuroscience, vol. 2016, Article ID 3289801, 11 pages, 2016. doi:10.1155/2016/3289801.

[3] Mohanty Sharada P., Hughes David P., Salathé Marcel,” Using Deep Learning for Image-Based Plant Disease Detection”, Frontiers in Plant Science, Volume 7, 2016.

[4] S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," *2015 International Conference on Computing Communication Control and Automation*, Pune, 2015, pp. 768-771.

[5] Vijai Singh, A.K. Misra, ”Detection of plant leaf diseases using image segmentation and soft computing techniques”, Information Processing in Agriculture, Volume 4, Issue 1,2017,Pages 41-49,ISSN 2214-3173.

[6] D. Al Bashish, M. Braik and S. Bani-Ahmad, "A framework for detection and classification of plant leaf and stem diseases," *2010 International Conference on Signal and Image Processing*, Chennai, 2010, pp. 113-118.

[7] S. K. Tichkule and D. H. Gawali, "Plant diseases detection using image processing techniques," *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, Coimbatore, 2016, pp. 1-6.

P:F-SMR-UG/08/R0

[8] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, “A review of advanced techniques for detecting plant diseases,” Computers and Electronics in Agriculture, vol. 72, no. 1, pp. 1–13, 2010.
[9] P. Chaudhary, A. K. Chaudhari, A. N. Cheeran, and S. Godara, “Color transform based approach for disease spot detection on plant leaf,” International Journal of Computer Science and Telecommunications, vol. 3, no. 6, pp. 65–69, 2012.
[10] M. S. P. Babu and B. Srinivasa Rao, “Leaves recognition using back propagation neural network-advice for pest and disease control on crops,” IndiaKisan. Net: Expert Advisory System, 2007.
[11]https://www.crowdai.org/challenges/plantvillage-disease-classification-challenge/dataset_files