2018 IEEE International Conference on Innovative Research and Development (ICIRD)
11-12 May 2018,Bangkok Thailand

Plant Disease Detection Using CNNs and GANs as an Augmentative Approach

Rutu Gandhi
Pune Institute of Computer Technology
Pune, India
rutugandhi21@gmail.com

Shubham Nimbalkar
Pune Institute of Computer Technology
Pune, India
shubhamn21@gmail.com

Nandita Yelamanchili
Pune Institute of Computer Technology
Pune, India
nndty28@gmail.com

Surabhi Ponkshe
Pune Institute of Computer Technology
Pune, India
surabhi.ponkshe@gmail.com

***Abstract*—Almost 40% of the world's crop yield is lost to diseases and pest infestations. According to a 2012 survey, Maharashtra has the highest rate of farmer suicides and one of the major reasons for this is the failure of crops. This paper presents an image-based classification system for identification of plant diseases. Since existing datasets have diluted focus across several countries and there are none that pertain to India specifically, there is a need for establishing a local dataset to be of use to Indian farmers. It uses Generative Adversarial Networks (GANs) to augment the limited number of local images available. The classification is done by a Convolutional Neural Network (CNN) model deployed in a smart phone app.**

Keywords—convolutional neural networks; generative adversarial networks; deep learning

I. INTRODUCTION

While India has progressed in a number of different sectors, the development of agriculture in India has not grown in parallel with the new available technologies. Diseases and pest infestations in agricultural crops are one of the major threats to food security. Plant diseases also pose a threat to the income of smallholder farmers. This paper presents an effort to create a system for the automatic image-based classification of plant diseases that can be deployed as a smartphone app.

The proposed system focuses only on agricultural crops grown on a large scale for the purposes of profit. As such, it can also be used by small scale growers of these plants. All the diseases are identified based on the image of a single leaf of the affected plant. While all these plants have fruit that show the disease as well, our aim is to detect the disease at an earlier stage, before it affects the fruit, so as to decrease losses. For this reason, the proposed system classifies plant species and diseases based only on the leaf. Some diseases also show symptoms on the stalk of the plant, but since these are less in number and they also simultaneously appear on the leaves, the system does not consider these symptoms.

The proposed system is based on Convolutional Neural Networks (CNNs), a popular deep learning technique used particularly for image classification. Two CNN architecture models – Inception v3 and MobileNets -- have been experimented with to compare both in terms of accuracy, speed of training and size of the mod el, among other factors. The system also makes use of Generative Adversarial Networks (GANs) to augment limited datasets of Indian plants and diseases. Details of all these techniques are elaborated on in later sections.

II. RELATED WORK

Krizhevsky et al. (2012)[3] showed for the first time that end-to-end supervised training using a deep convolutional neural network architecture is a practical possibility even for image classification problems with a very large number of classes, beating the traditional approaches using hand-engineered features by a substantial margin in standard benchmarks. This is a promising candidate for a scalable and practical approach for computational inference of plant diseases due to the absence of the labor-intensive phase of feature engineering and the generalizability of the solution.

Also, this approach is based on the work by S.P. Mohanty et al. (2016)[1] that focuses on the comparison between Alexnet and GoogleNet along with other parameters like the splitting percentage of the training and testing data, segmented/unsegmented leaf images, grey-scale/coloured dataset and transfer learning/training from scratch. All combinations of the above parameters have been studied in this paper. The best models for the dataset were GoogLeNet:Segmented:TransferLearning:80–20 and GoogLeNet:Color:TransferLearning:80–20.

Importantly, while the training of the model takes a lot of time (multiple hours on a high performance GPU cluster computer), the classification itself is very fast (seconds on a CPU), and can thus easily be implemented on a smart phone.

III. DATASET

Although Indian agriculture researchers lack contribution of a large dataset of leaves, at an international level, the Plant

978-1-5386-5696-9/18/$31.00 ©2018 IEEE

Village from Pennsylvania State University is a well known dataset for this research. It contributes solutions to more than 152 crops[4]. A dataset of 38 classes (19 crops) has been used, which amounts to more than 56,000 images from the same source. Collection of acquired dataset includes crops grown locally along with classes of diseases that those crops can have. The dataset represents high quality images of leaves. Each image is of jpeg format with width: 5472 pixels and height: 3648 pixels. After preprocessing of the dataset to make it noise free and applying segmentation, the size of dataset was reduced significantly. The preprocessed dataset comprised of images of width: 256 pixels and height: 256 pixels. The various models created were based upon either segmented dataset or with a background. This was used to check which would give better accuracy.

Creating a mask to eliminate the underlying background was based upon eradicating the blue channel. Since the color blue rarely coincides with the leaf color or the disease, the mask helps in retaining the subject.


Cropped Image


Segmented Image

IV. Generative Adversarial Networks

Since all the related work has a global scope with diluted focus on Indian crops and diseases, the aim here is to put local agriculture on priority. Even so, the availability of such a native image dataset for training the model is limited. A lot more (and more variable) data is required to train the model so that it achieves considerable accuracy. In this approach, Generative Adversarial Networks play an important role in achieving that. These networks are being used in augmenting the available limited data for local plants and diseases. GANs have shown promising results in generating images [4][3] with significant detail.

However, GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs. Thus, we have used a class of GANs called deep convolutional generative adversarial networks (DCGAN)[2],that have certain architectural constraints, and are a strong candidate for unsupervised learning. DCGANs are a type of GAN whose architecture has been modified to result in more stable training for a broad range of datasets that create deeper generative models and higher resolution pictures.

Reference [2]'s implementation of the DCGAN architecture was trained on the class of segmented healthy apple leaves in the PlantVillage dataset on an NVIDIA GPU over 900 epochs. This resulted in the generator rendering extremely realistic images that were able to fool a normal CNN classifier trained on the same dataset.


Epoch 3 Epoch 27 Epoch 899


Machine Generated Image


Real Leaf

V. Approach

The main classifier of the system is a Convolutional Neural Network (CNN) [3][11], which is a network, made up of different types of layers of neurons or perceptrons. All the layers work towards extracting the most important features in the images of the dataset. The layers are as follows:

- Convolution Layer:
  This layer performs a convolution operation of a weight matrix (or filter) with the input image to produce a stack of filtered images. The filter is multiplied with patches of the image matrix chosen over a particular stride.
- Pooling Layer:
  Pooling layer is responsible for reducing the number of parameters in the image stack, and in turn, reducing the amount of computation required. The most common form of pooling is MaxPooling, where from every small pooling window, the maximum value is selected.
- Activation Layer:
  Activation layer is used to monitor the firing rate of the neurons i.e. apply non-linearity to the input. Commonly used activation functions are:

978-1-5386-5696-9/18/$31.00 ©2018 IEEE

- *ReLU (Rectified Linear Units:* [10] Non-linearity is thresholded at 0.

$$f(x) = \max(0, x)$$

- *Sigmoid:* It fits the range of the input between 0 and 1.

$$\sigma(x) = 1/(1 + e^{-x})$$

- *Tanh:* Input range is fitted between -1 and 1.

$$Tanh(x) = 2\sigma(2x) - 1$$

- Softmax Classifier:
  This is used in the final layer of CNNs to give probability values for each class label. The softmax function is as follows [12]:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{k} e^{x_j}}, i = 0,1,2, \ldots k$$

A. *INCEPTION v3*

This version of the Inception model (Szegedy et. al)[5] builds further on the idea of the model choosing between multiple convolutions (1x1, 3x3, 5x5, 7x7). However, it makes several improvements in terms of the computational expense of running the model. Larger spatial convolutions are more computationally expensive, making the model slower and heavier to train as a whole. Inception v3 replaces such larger convolutions with a network of smaller convolutions. 5x5 convolutions are replaced with a two-layer architecture: one 3x3 convolution followed by a fully connected layer on top of the 3x3 output grid generated by the previous 3x3 layer. This essentially comes down to all 5x5 convolutions being replaced by two 3x3 convolutions. 3x3 layers are also replaced with two convolutions: a 3x1 convolution followed by a 1x3 convolution, resulting in 33% cheaper computation. 7x7 layers are also replaced with 7x1 and 1x7 convolutions.

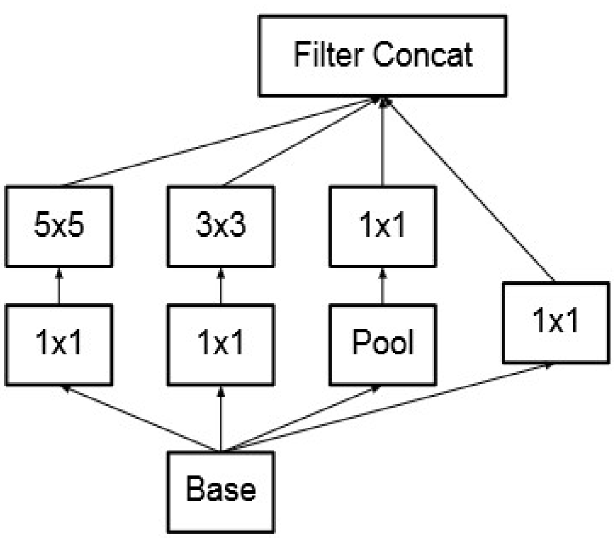

Fig: Original Inception model architecture

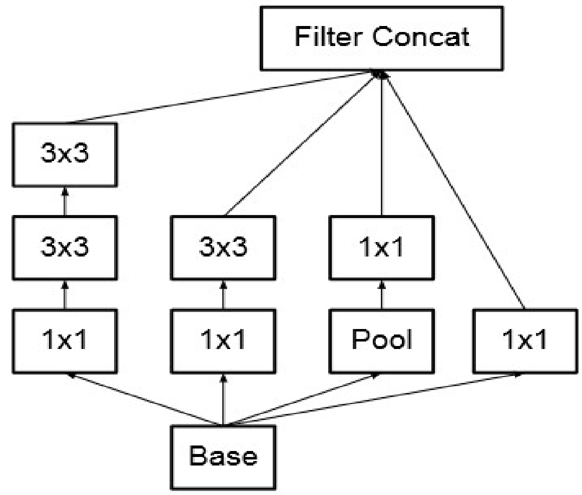

Fig: Modified Inception architecture (v3) with 5x5 convolutions replaced with 2 3x3 layers

For the proposed system, an Inception v3 model was pre-trained on the ImageNet database, and retrained the final layer to work with the dataset of plant diseases. Training was performed over 4000 steps, with a 75:5:20 training, testing and validation ratio of the unsegmented image data. Learning rate was set to 0.01. This model achieved an accuracy of 88.6%.

B. *MOBILENETS*

While accuracy achieved by the Inception models are high, they are still heavier to run and slower to compute, making it necessary for the classifier to be run as a web service in the cloud and have a mobile device communicate with that service. To counter this drawback, we experiment with the usage of the 28-layer MobileNets architecture [6] of CNNs. The core of the Mobilenet architecture is depth wise separable convolutions. It is a kind of factorized convolution where a normal convolution operation is factored into a depthwise convolution followed by a 1x1 pointwise convolution. In a depthwise convolution, a single filter is applied to each input channel, whereas in point wise convolutions, the outputs of the depth wise convolutions are combined. This two-step convolution method drastically reduces computation time and size of the model as opposed to standard convolution where filtering and combining outputs is all done in a single step.

The first layer is a full convolution, followed by depth wise separable convolutions. Every layer is followed by a batch normalization layer and ReLU nonlinearity. The final fully

978-1-5386-5696-9/18/$31.00 ©2018 IEEE

connected layer is the only one with no non-linearity. It is instead followed by a softmax layer which performs the final classification. The fully-connected layer is also preceded by an average pooling layer to reduce the spatial resolution to 1.

The Mobilenets model used for plant disease detection has been pretrained on the ImageNet dataset, with the final layer retrained to work with the plant disease dataset. Training was performed over 4000 steps, with a 75:5:20 training, testing, validation ratio of the dataset of segmented images. Learning rate was set to 0.01. This model achieved a final test accuracy of 92%.

C. *Mobile Application*

The trained model was embedded inside a mobile application. When the camera is hovered over the leaf, the app takes the input, classifies it and displays prediction in real-time. The classifier runs on it and gives an output in the form of probabilities in the range [0,1]. Both the models - inception as well as the mobilenet – were deployed. Although the inception model works well on a computer system, its size is large (approximately 87 MB) and would take up a lot of memory and processing power on a mobile device. On the other hand, the mobilenet model is just 5.6 MB in size and the processing is much faster.

Since the app involves the use of a pre-trained model and is lightweight, it runs completely offline.

VI. Future Scope

Currently, the model is deployed on a mobile app and requires the farmer to physically move through the field and capture pictures. The future scope could be to employ unsupervised drones that navigate the field using computer vision similar to [7]. The drone would capture several pictures which could be sent back to the server on which the model runs giving a classification output.

There has also been significant research [8][9] in the modification of GANs for the discriminator to act as a multi-class classifier, so that it would have N+1 output labels as opposed to just two (real, fake). This approach could also be looked into for the plant disease detection problem. Further work in improving battery optimization for the smartphone app is also a necessary area of further research.

978-1-5386-5696-9/18/$31.00 ©2018 IEEE

REFERENCES

[1] S.P. Mohanty, D. P. Hughes, Marcel Salathe, "Using Deep Learning for Image-based Plant Disease Detection" (2016) Front. Plant Sci., arXiv: 1604.03169, 22 September 2016.

[2] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks" arXiv preprint arXiv:1511.06434, 2015.

[3] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, eds F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc.), 1097–1105.

[4] D. P. Hughes, M. Salathe, "An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics", CoRR abs/1511.08060.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Woina, "Rethinking the Inception Architecture for Computer Vision" arXiv:1512.00567v3, Dec 2015

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" arXiv:1704.04861, April 2017

[7] Alessandro Giusti et. al "A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots" IEEE Robotics and Automation Letters, Preprint version accepted November, 2015.

[8] Augustus Odena. "Semi-supervised learning with generative adversarial networks". arXiv preprint arXiv:1606.01583, 2016.

[9] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. "Improved Techniques for Training GANs". ArXiv e-prints, June 2016.

[10] Vinod Nair, Geoffrey E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", ICML, 2010.

[11] Y. LeCun, P. Haffner, L. Bottou, Y. Bengio, "Object Recognition with Gradient-Based Learning", *Feature Grouping*, 1999.

[12] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo. "Multi-category classification by soft-max combination of binary classifiers." In Multiple Classifier Systems (MCS), pages 125–134, 2003

978-1-5386-5696-9/18/$31.00 ©2018 IEEE