Computer Vision using Machine Learning

Assignment 3

IMAGE BASED PLANT DISEASE DETECTION

Abhishek Kumar	150101003
Gourav	150101024
Mayank Agrawal	150101033
Saswata De	150101058
Vaibhav Singh	150101081

Abstract

The latest generation of convolutional neural networks (CNNs) has achieved impressive results in the field of Image classification. We trained a large, deep convolutional neural network to classify 20,639 images of diseased plants. On the test data we have achieved 96.77 percent of accuracy. The neural network consists of 5 convolutional layer, some of which are followed by a max-pooling layer. In total 3 max-pooling layers followed by 2 dense layers with a final 15-way softmax layer. For overfitting reduction dropout method has been used effectively.

Introduction

Agricultural productivity is something on which economy highly depends. This is the one of the reasons that disease detection in plants plays an important role in agriculture sector, as getting disease in plants are quite natural. If proper care is not taken in this regard then it can cause serious effects on plants and due to which respective product quality, quantity and productivity is adversely affected.

Implementing the appropriate management strategies like fungicide applications, disease-specific chemical applications, and vector control through pesticide applications could lead to early information on crop health and disease detection. This could facilitate the control of diseases and improve productivity.

A proper disease detection will ensure the well being of plants. Experts on plant disease can detect diseases by seeing the plants. But a farmer or a normal gardener can't detect these symptoms by bare eyes. Our implementation helps in identifying the plant type and respective disease caught by capturing the image of leafs. For example black rot or scab in apple leafs.

Proposed Method

To solve this problem, we used AlexNet. AlexNet is a convolutional neural network proposed by Alex Krizhevsky on 2012. AlexNet contained eight layers; the first five were convolutional layers, three of them followed by max-pooling layers, and the last three were fully connected dense layers. It used the non-saturating ReLU activation function, which showed improved training performance over tanh and sigmoid.

Using a public dataset of 20639 images of diseased and healthy plant leaves collected under controlled conditions, we train a deep convolutional neural network to identify 3 crop species and 15 diseases (or absence thereof).

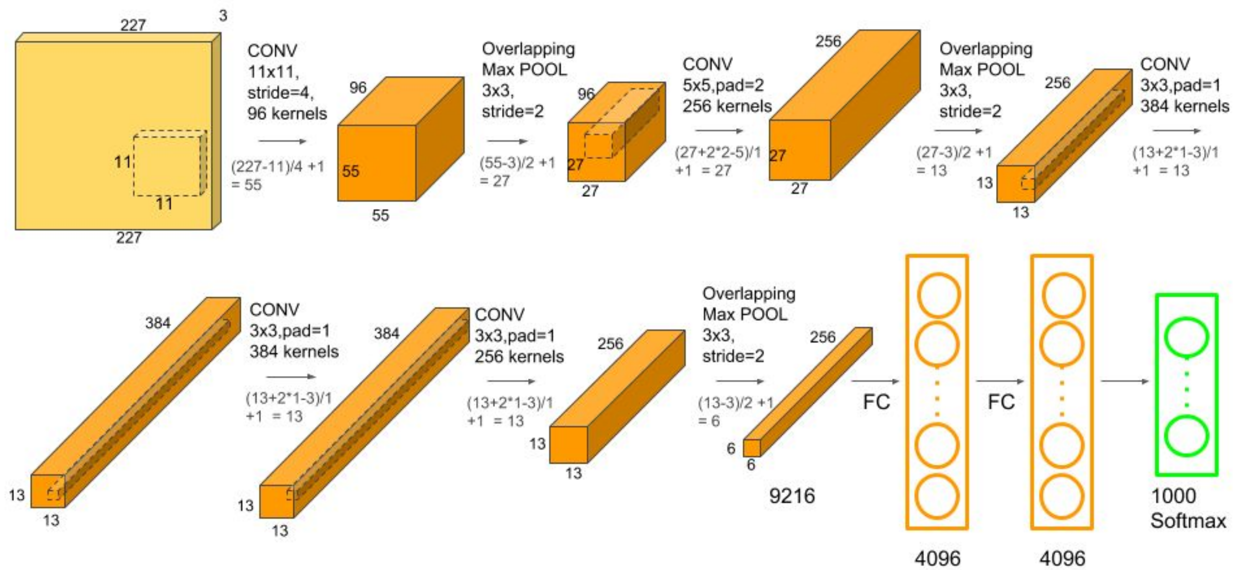
Pre-processing methods used

Building an effective neural network model requires careful consideration of the network architecture as well as the input data format. We need to remove the differences in the images due to different aspect ratios, intensity differences and background clutter for better performance. Some of the precaution steps taken are:

- Maintain uniform aspect ratio and scaling
- Remove background of leaf images
- Zoom on input image
- Dimensionality reduction
- Image augmentation
- Normalizing image input

AlexNet Architecture

Let's look at the architecture of Alexnet. *(Sample image of used architecture)*



AlexNet consists of 5 Convolutional Layers and 3 Fully Connected Layers. Various Convolutional Kernels, also known as filters extract several interesting features from an image. We usually use several kernels of the same size in a single convolutional layer. For example, the first convolution Layer of AlexNet contains 96 kernels of size 11x11x3. The width and height of the kernel signifies size of convolutional matrix and is same for all kernel in a single layer and the depth signifies the number of channels, for the first layer, which is 3 (for each of red,green & blue).

The 1st & 2nd convolutional layers of Alexnet are followed by a Max Pooling layers. The 3rd, 4th and 5th convolutional layers are connected directly. The 5th convolutional layer is followed by an Overlapping Max Pooling layer, the output of which goes into a series of two fully connected ense layers. The second fully connected layer goes into a softmax classifier with class label equal to no of different classes, for example, in our case, 15.

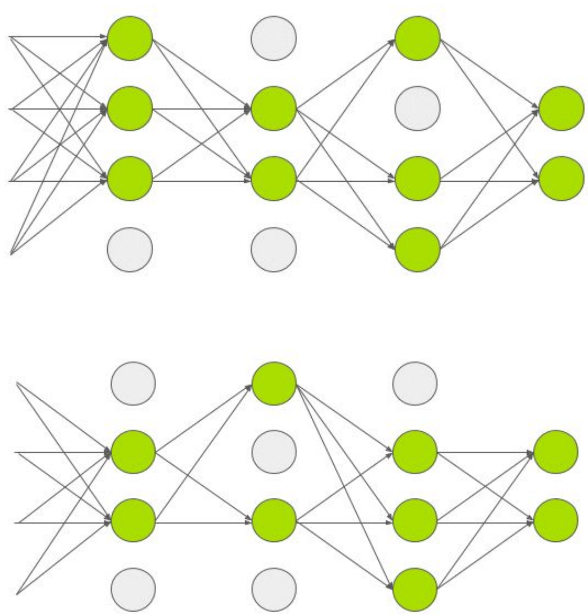
ReLU nonlinearity is applied after all the convolution and fully connected layers. The ReLU nonlinearity of the first and second convolution layers are followed by a local normalization step before doing pooling.

Reducing Overfitting

The size of the Neural Network specify how much it can learn, but if we don’t take care, it will try to remember the examples in the training data without understanding the concept. Thus the Neural Network will work very well on the training data, but they will fail to learn the underlying concept, and thus It will not work well on new and unseen test data. This is called overfitting.

Dropout

In dropout, a neuron is dropped from the network with a probability of 0.5. A dropped neuron will not contribute to forward or back-propagation. So the input will go through a different neural net architecture. Thus, the learnt parameters become more robust and do not get overfitted. During testing, the whole network, including dropped neurons, is used, and output is scaled down by 0.5 to compensate the missed neurons while training. Dropout increases epochs, the number of iterations needed to converge, by a factor of 2, but without dropout, AlexNet would overfit substantially.

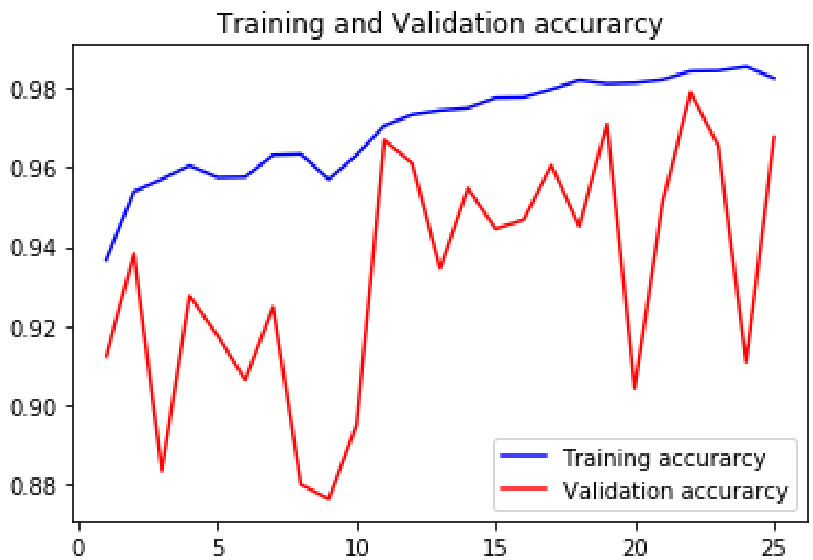


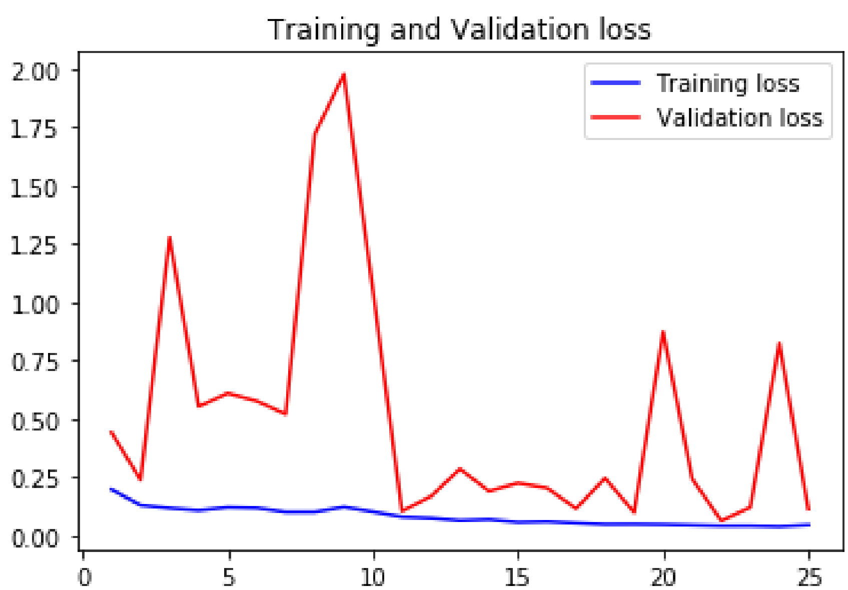
Summary of our Alexnet model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 256, 256, 32)      896
_________________________________________________________________
activation_1 (Activation)    (None, 256, 256, 32)      0
_________________________________________________________________
batch_normalization_1 (Batch (None, 256, 256, 32)      128
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 85, 85, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 85, 85, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 85, 85, 64)        18496
_________________________________________________________________
activation_2 (Activation)    (None, 85, 85, 64)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 85, 85, 64)        256
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 85, 85, 64)        36928
_________________________________________________________________
activation_3 (Activation)    (None, 85, 85, 64)        0
_________________________________________________________________
batch_normalization_3 (Batch (None, 85, 85, 64)        256
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 42, 42, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 42, 42, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 42, 42, 128)       73856
_________________________________________________________________
activation_4 (Activation)    (None, 42, 42, 128)       0
_________________________________________________________________
batch_normalization_4 (Batch (None, 42, 42, 128)       512
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 42, 42, 128)       147584
_________________________________________________________________
activation_5 (Activation)    (None, 42, 42, 128)       0
_________________________________________________________________
batch_normalization_5 (Batch (None, 42, 42, 128)       512
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 21, 21, 128)       0
_________________________________________________________________
dropout_3 (Dropout)          (None, 21, 21, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 56448)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              57803776
_________________________________________________________________
activation_6 (Activation)    (None, 1024)              0
_________________________________________________________________
batch_normalization_6 (Batch (None, 1024)              4096
_________________________________________________________________
dropout_4 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 15)                15375
_________________________________________________________________
activation_7 (Activation)    (None, 15)                0
=================================================================
Total params: 58,102,671
Trainable params: 58,099,791
Non-trainable params: 2,880
_________________________________________________________________
```

Result

We were able to achieve a accuracy of 96.77% on our validation set.





Conclusion

In this project, a new approach of using deep learning method to automatically classify and detect plant diseases from leaf images was explored. The developed model was able to detect leaf presence and distinguish between 15 different diseases, which may not be visually diagnosed by bare un-trained eyes. An extension of this study will be on gathering images for enriching the database and improving accuracy of the model using different techniques of fine-tuning, augmentation and boosting.

The main goal for the future work will be on developing a complete system consisting of server side components containing a trained model on this regard and an application for smart phones with features such as displaying recognized diseases in fruits, vegetables, and other plants, based on leaf images captured by the mobile phone camera. This application will serve as an aid to farmers (regardless of the level of experience), enabling fast and efficient recognition of plant diseases and facilitating the decision-making process when it comes to the use of chemical pesticides. The accuracy of detection may vary depending on the quality of image, surrounding lighting condition and the level of pixel details.