Slide 1:

What Is Transfer Learning?

Transfer learning generally refers to a process where a model trained on one problem is used in some way on a second related problem.

Slide 2:

For this challenge, we used the “[PlanVillage](https://plantvillage.psu.edu/)” dataset. This dataset contains an open access repository of images on plant health to enable the development of mobile disease diagnostics. The dataset contains 54, 309 images. The images span icludes 14 crop species: Apple, Blueberry, Cherry, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato. Of these crops we have these classes which are 17 fundal diseases, 4 bacterial diseases, 2 molds (oomycete) diseases, 2 viral diseases, and 1 disease caused by a mite. 12 crop species also have images of healthy leaves that are not visibly affected by a disease.

Slide 3:

In this work, we study transfer learning of the deep convolutional neural networks for the identification of plant leaf diseases and consider using the pre-trained model learned from the typical massive datasets(imagenet), and then transfer to the specific task trained by our data.

Slide 4:

The majority of plant diseases show visible symptoms.

For a plant pathologist to accurately diagnose plant disease, he must have good observation skills and thus identify characteristic symptoms. However, the excessive variety of plants, variations in plant diseases due to climate changes, and the faster spread of diseases to other regions where they have not been before even lead experienced pathologists to fail to diagnose certain diseases.

The presence of expert and intelligent systems that can automatically diagnose plant disease accurately provides valuable contributions to farmers.

Slide 5:

In the past approaches, we mostly had to extract the features ourselves to enhance the classification accuracy using Different feature extraction techniques. In neural networks this process of feature extraction can be automated. However, in normal neural networks, our accuracy can be deterred by small variations in the picture taken of the sample.

All of this has prompted us to come up with new techniques for better and faster performance.

Slide 6:

Slide 7:

Slide 8:

Slide 9:

Pre-processing

At first, we load the dataset using Keras preprocessing package, splitting it into two sets of training 80% and validation 20%. We check the number of classes and their names and plot some of them with their labels to visually make sure there is no error in either part. Then we proceed to data augmentation and preprocessing. We again take 20% of the validation set to create a test dataset by taking one of every five pictures out of the set. We augment images after that by randomly flipping and rotating them as shown in Figure.