Plant Disease Detection using a CNN framework and Transfer Learning

Mathew Shaham

EE428 Computer Vision

12/11/2023

Previous and Other Interesting Work

- Crop Leaf Disease Recognition Based on Self-Attention Convolutional Neural Network
- PDDD-Pretrain: A series of Commonly Used Pre-Trained Models Support Image-Based Plant Disease Diagnosis
- Explainable Vision Transformer Enabled Convolutional neural Network for Plant Disease Identification: PlantXVIT
- VGG-ICNN (lightweight CNN on the right, ImageNet Transfer Learning for VGG16-Network)
- And More! (Tranvolution with GAN - preGAN / postGAN / Feature Pyramid Network Backbone)

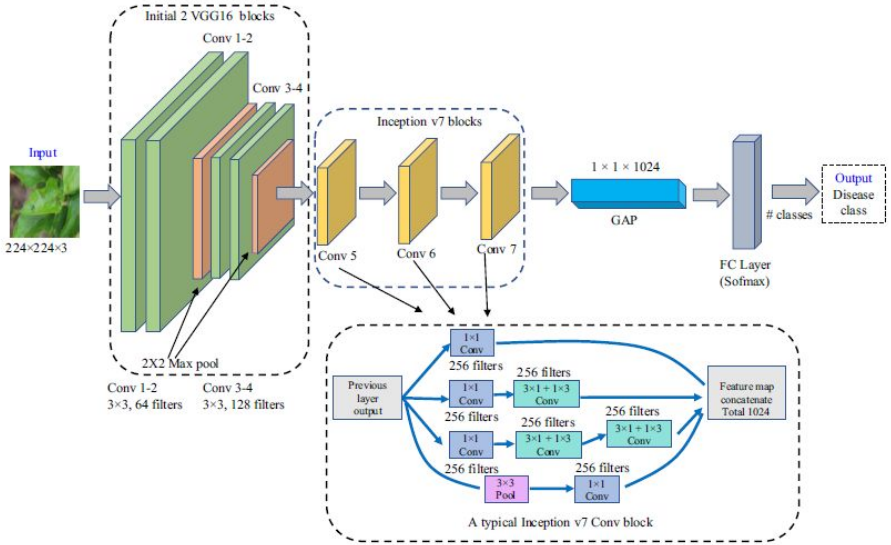
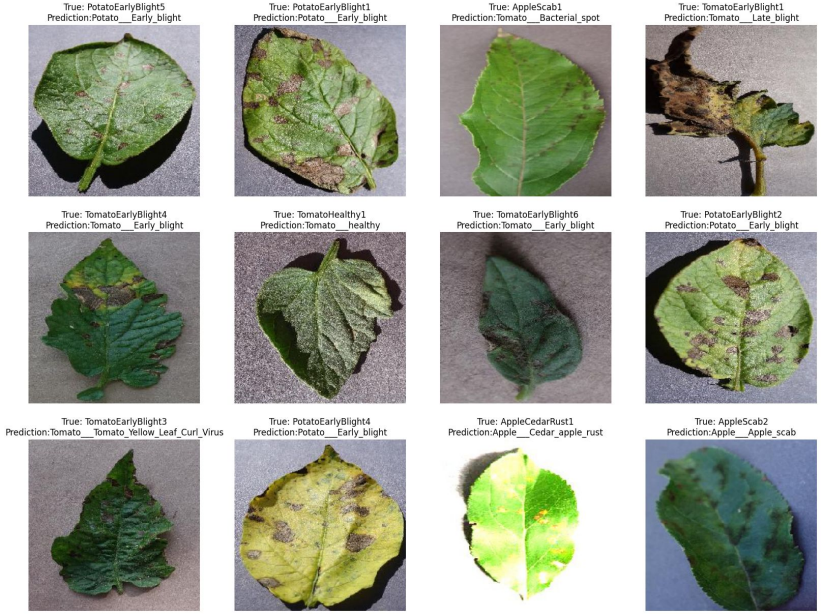


Fig. 1 VGG-ICNN structural diagram

Dataset Comparison

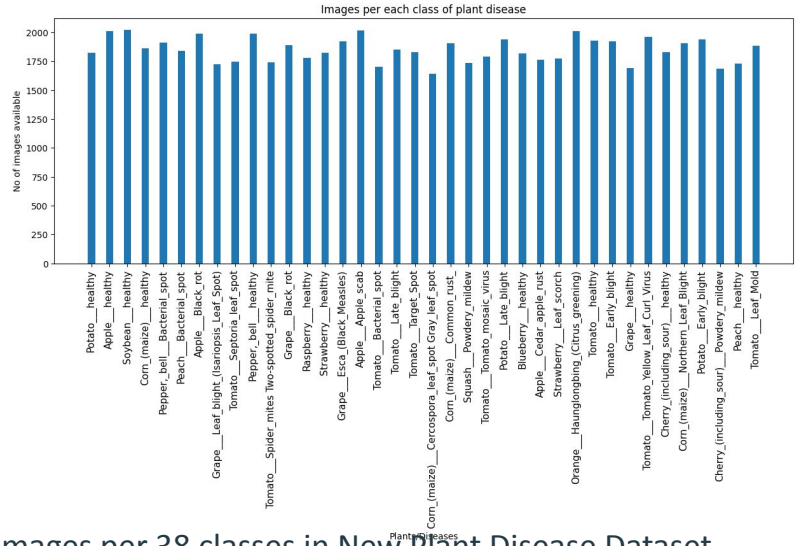
	Number Images / Classes	Pros	Cons / Notes
PlantVillage	54,303 / 38 classes	Original / Widely Used	Only Lab environment (background not realistic)
PlantDoc	2,569 / 30 classes (13 plants)	More Realistic Images	Read some images are labeled wrong
FieldPlant	5,170 / 27 classes (8,629 annotated leaves)	More Realistic Images	-
Embrapa	2326 / 171 classes - diseases (21 plants)	Large # Classes	Images subdivided 46,513 images total
AI Challenger 2018	61 classes	More Classes	Some Classes have few images (less than 10)
New Plant Disease Dataset	38 classes		Just Modified Plant Village

Dataset Sample Images



Example after classification / prediction

Note some of the augmentation on images

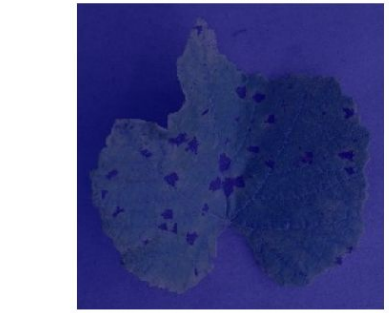


images per 38 classes in New Plant Disease Dataset

PlantDoc: A Dataset for Visual Plant Disease Detection:
https://paperswithcode.com/paper/plantdoc-a-dataset-for-visual-plant-disease

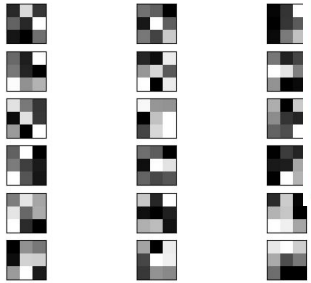
My Work (1)

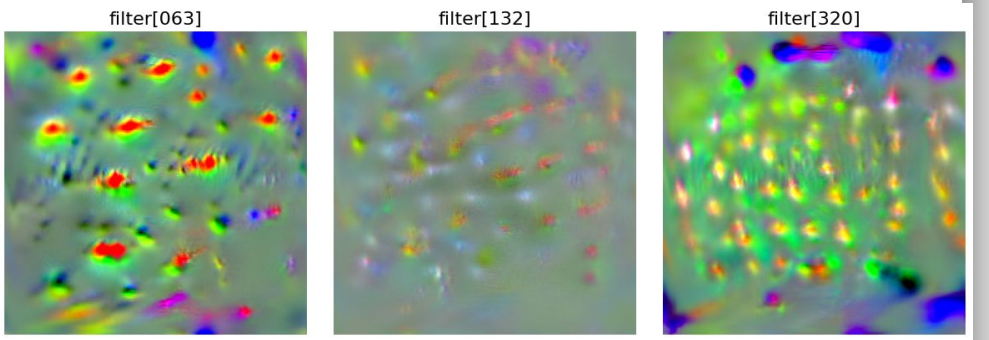
- Saliency and GradCAM
  - Use linear output instead of softmax to create heatmap on output to
  - visualize the gradients of a specific layer within the network
  - Activation Maximization: Initialize Pixels Randomly - maximize each neuron for given set filter
- Viewing Convolution FIlters - from last (6th) layer of CNN
- Class Activation Maps
- Saliency maps: aim to define where a particular region is different / noticeable from its neighbors with respect to image features



GradCAM (attempt)

Source: https://keisen.github.io/tf-keras-vis-docs/examples/attentions.html

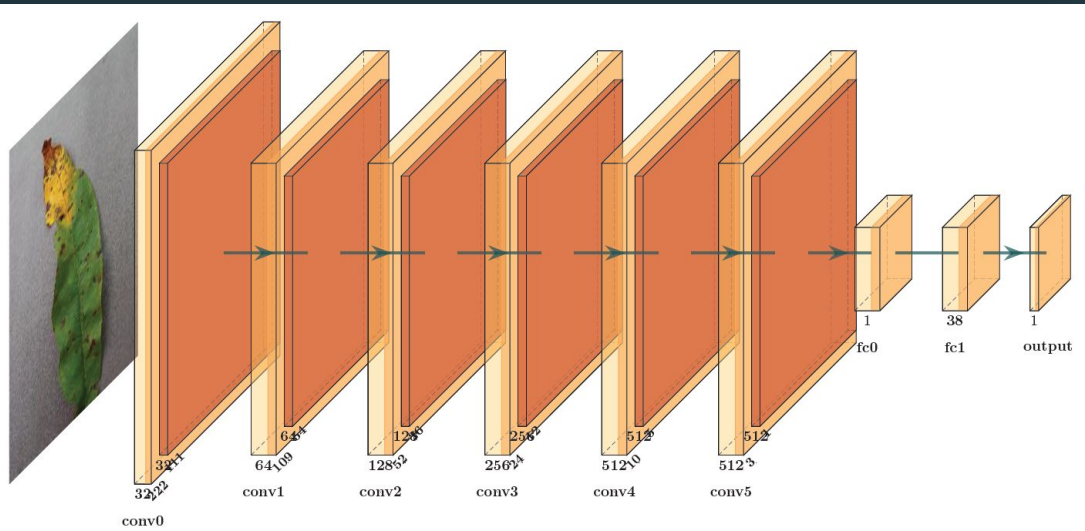




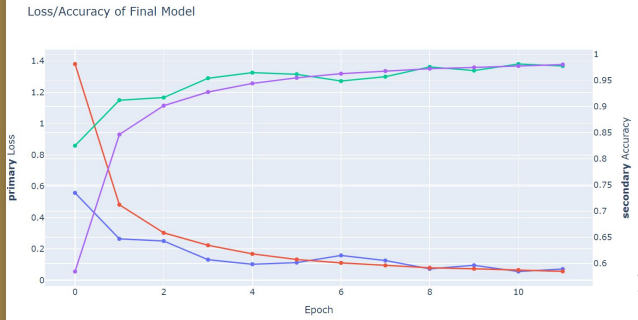
Normalized Convolutions

My Work (2)

- Changing the number of outputs and training the network on a different dataset
- Would have like to / might try a 3rd dataset (more realistic one)
- Training times around 200s per epoch for Transfer Learning (61 classes, 5 epochs, 25 batch size)



Network Used




```
Epoch 1/5
1310/1310 [==============================] - 245s 175ms/step - loss: 0.9155 - accuracy: 0.7148 - val_loss: 0.5024 - val_accuracy: 0.7999
Epoch 2/5
1310/1310 [==============================] - 189s 144ms/step - loss: 0.5000 - accuracy: 0.8079 - val_loss: 0.5016 - val_accuracy: 0.8051
Epoch 3/5
1310/1310 [==============================] - 182s 139ms/step - loss: 0.4333 - accuracy: 0.8286 - val_loss: 0.5214 - val_accuracy: 0.7983
Epoch 4/5
1310/1310 [==============================] - 178s 136ms/step - loss: 0.3925 - accuracy: 0.8405 - val_loss: 0.4798 - val_accuracy: 0.8033
Epoch 5/5
1310/1310 [==============================] - 180s 137ms/step - loss: 0.3602 - accuracy: 0.8528 - val_loss: 0.4868 - val_accuracy: 0.8005
```

12 epochs batch size of 50 performance

Future Improvement

- Detecting exact locations and types of diseases (Mask-RCNN)
- Providing feedback on potential causes and remedies for the disease
  - Actual PlantDoc
- Speed Improvements (other pre-trained models)
- Smaller Models (for deployment in phones)
- Collecting more data / creating datasets
  - Embrapa - large dataset/#classes but few images
  - Ensuring data in lab environment and in actual environment
- Increased complexity in training (rotation, brightness, other image pre-processing)

Source: https://github.com/AarohiSingla/Plant-Disease-Detection-Using-Mask-R-CNN/tree/main