Final Project

Identify Tree Species from Leaf Images

Trias, Fernando

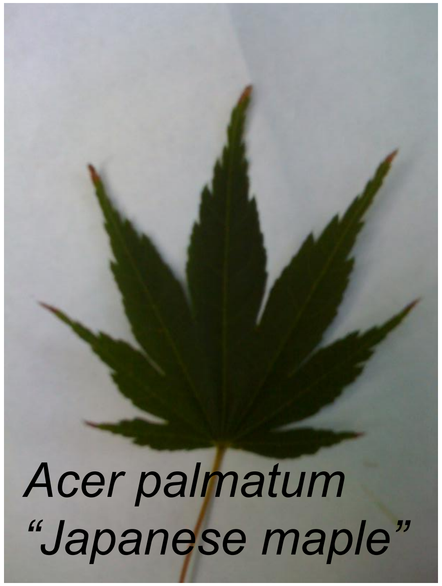
CSCI E-89 Deep Learning

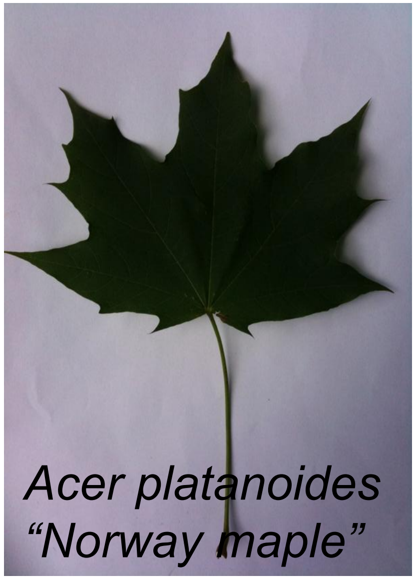
Harvard University Extension School

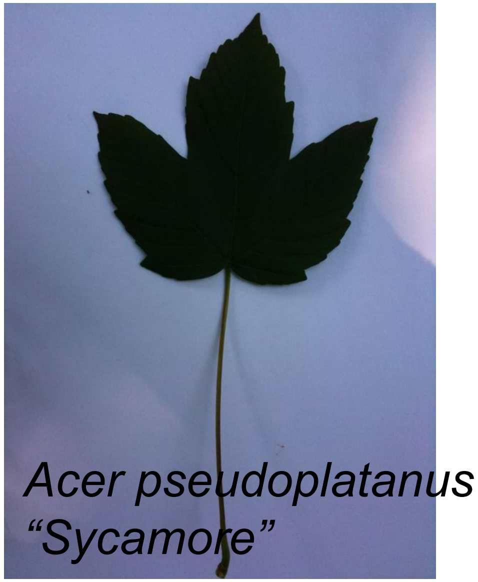
Prof. Zoran B. Djordjević

Introduction

- For broad-leaf trees, it is often (but not always) possible for an expert to identify the species of tree by merely observing the leaves. However, it is difficult for non-experts to do so.
- Computers generally have had unsatisfactory results. Many existing systems use image processing to identify properties of the leaves, like branching patterns, points and shape, and then traverse a decision tree, but this method is unreliable.
- I wanted to investigate if a convolutional neural network could identify the species of a tree by only looking at an image of a leaf.







Dataset

- Leafsnap is a collaboration by researchers from Columbia University, the University of Maryland, and the Smithsonian Institution.
- They provide an App that identifies Northeastern trees from an image of a leaf. Their App uses image processing to identify features and then identify a species using a decision tree.
- In addition, they provide all the raw data used to test their system at http://leafsnap.com/static/dataset/leafsnap-dataset.tar. This data is 976 MB. However, after removing pine trees and other non-broadleaf trees, the images are 334 MB.

Software & Hardware

- Amazon Linux (p2.xlarge)
- Keras
- Tensorflow
- OpenCV

Cleaning and Preprocessing

- Only consider trees in the Northeastern US for which the data is of high quality
- Removing non-broadleaf trees (pine trees, spruces, etc) leaves this data.
  - 26603 images
  - 159 tree species
- Converting to grayscale
  - Although the color of the leaf is helpful in identifying a tree, the images are not taken under consistent lighting and thus the colors are not very accurate.
  - Analyzing grayscale will speed up training.
- Resolution of 224x224
  - Features such as veins and leaf edges are small and so the image must be large enough to maintain some of these features.
  - When scaling images, aspect ratio must be maintained because overall shape is important. Thus, a custom scaling function is employed.
- Augmentation
  - Rotation, width/height shifts, zoom and small shear

Train, Validation and Testing

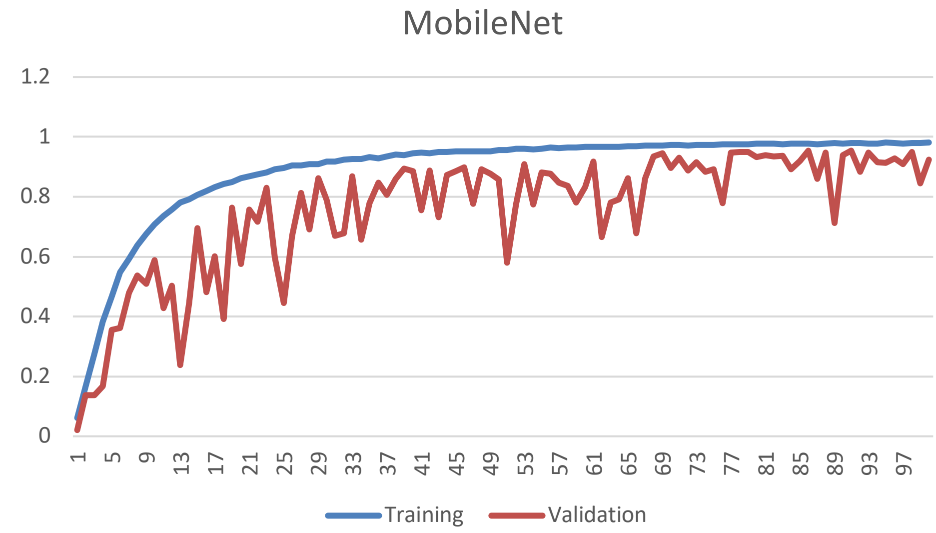
- Data split into three non-overlapping groups: Train, Validation and Testing. Each group must have several samples of each tree species.

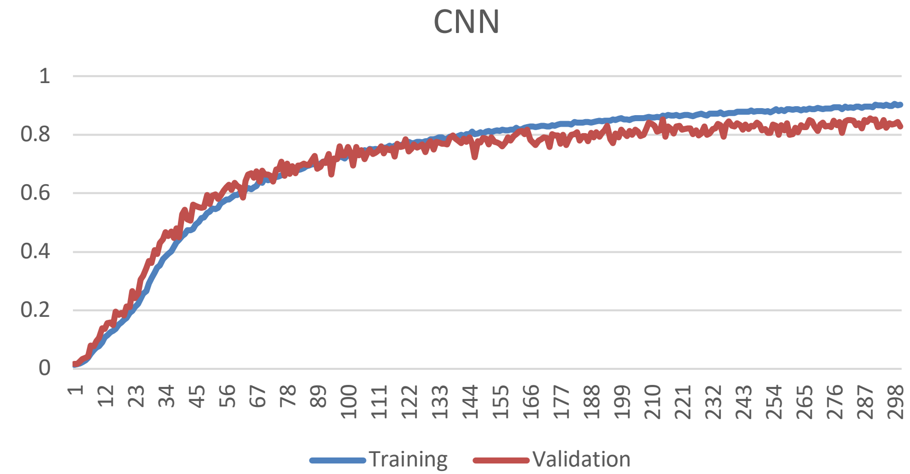
Group	Images	Percent
Training	21285	80%
Validation	2661	10%
Testing	2657	10%

- Training set will be used to train the neural network
- Validation set will be used during the training for validation at the end of each epoch.
- Testing will be used after the model has been trained to evaluate and compare with other models.

Models Tried

- MobileNet
  - Provided by Keras
  - Trained for 100 epochs
  - SGD optimizer
    - Learning rate 0.01
    - Momentum 0.9
    - Decay 0.001/50
  - 2.5 hours run time
- Convolutional
  - 3 Convolutional layers
  - Trained for 300 epochs
  - SGD optimizer
    - Learning rate 0.01
    - Momentum 0.9
    - Decay 0.001/50
  - 16.5 hours run time





Comparison

- Evaluating the model over the Testing set shows that MobileNet is superior and that the validation accuracy is a good proxy for the model's results.
- Errors in identification were often due to bad quality of the original image, or images that failed to capture distinguishing features.

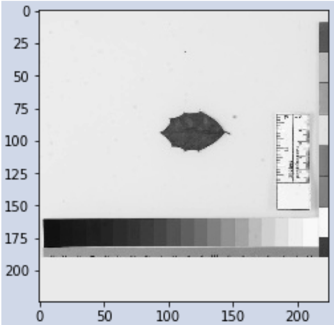
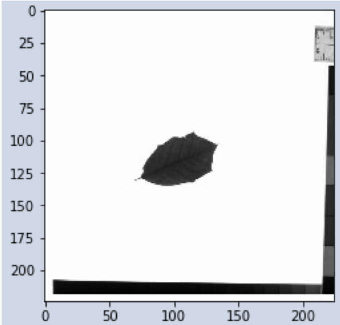
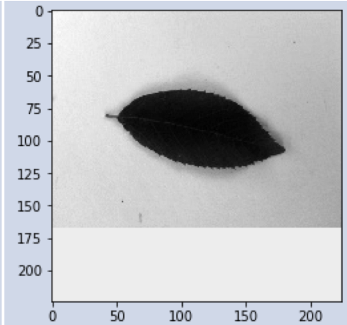
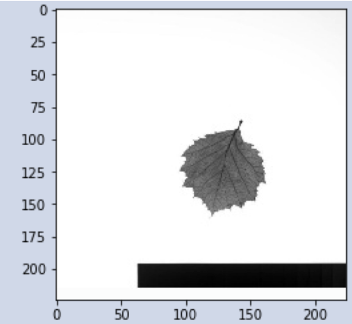
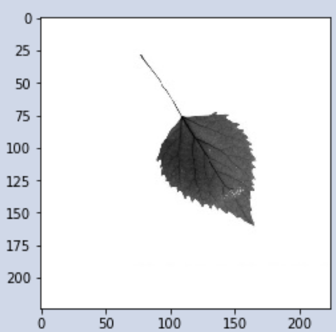
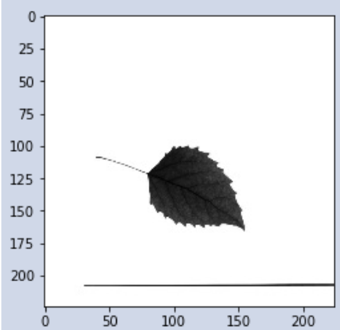
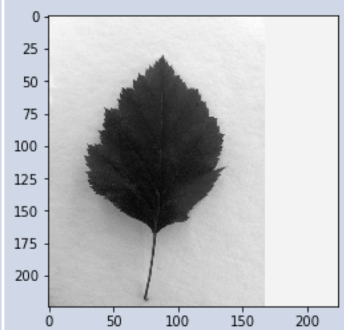
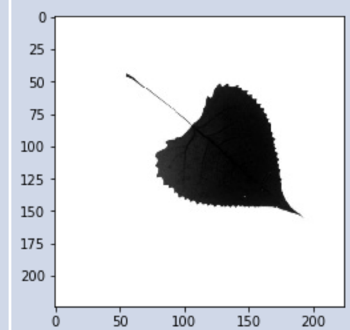
	MobileNet	CNN
Trainable parameters	3,369,375	13,225,119
Epochs	100	300
Training accuracy	0.9814	0.9016
Validation accuracy	0.9248	0.8290
Test accuracy	0.90625	0.8438
Training time	2.52 hours	16.47 hours

MobileNet

- Similar to ResNet and Inception
- Optimized for slow processors and limited memory
- Can also run without floating point calculations
- Supported by Tensorflow Lite and CoreML so it is easier to create mobile app.

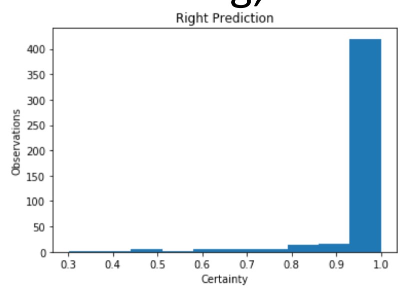
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

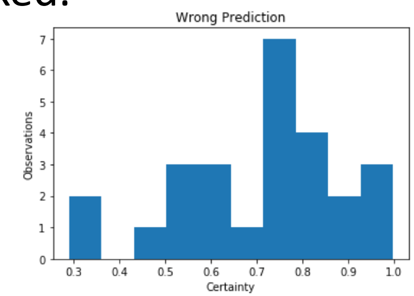
Examples

Actual	Top prediciton	2^{nd} prediction	3^{rd} prediction
			
ilex_opaca	0.99916005 ilex_opaca	0.00051971286 ostrya_virginiana	0.00011364489 platanus_occidentalis
			
populus_grandidentata	0.99978584 populus_grandidentata	0.00012750519 crataegus_pruinosa	3.2079985e-05 populus_deltoides

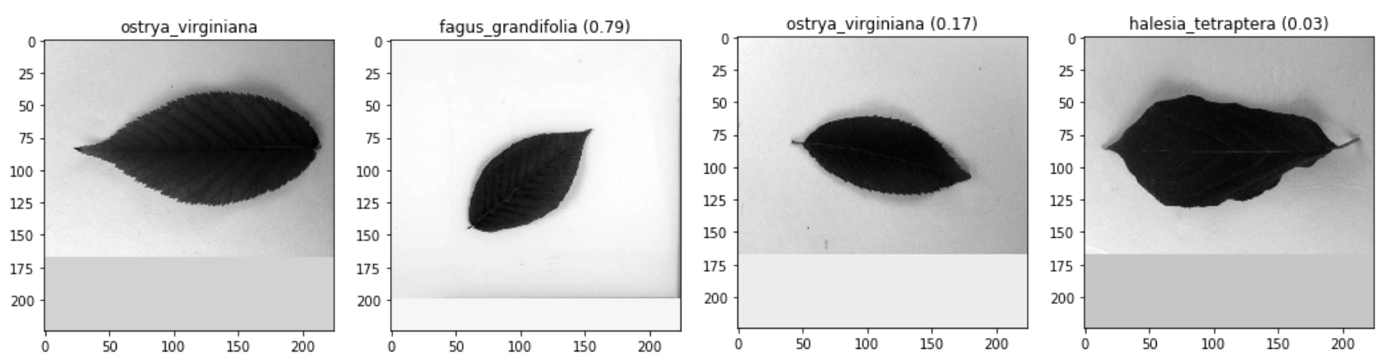
Certainty

- As shown below, when prediction is correct, certainty is very high.
- When prediction is wrong, certainty is mixed.





- This is because some leaves are indistinguishable. For example, see prediction for Ostoya virginiana below.
- Fagus grandifolia and Ostrya virginiana have virtually identical leaf images.
- But trees have very different size, shape and bark.



YouTube URLs, Last Page

- Two minute (short): https://youtu.be/S6jOuE9KHlg
- 15 minutes (long): https://youtu.be/TKbMa8bZzLQ
- Github: https://github.com/ftrias/leafident
- References

  - "Leafsnap: A Computer Vision System for Automatic Plant Species Identification," N Kumar, P Belhumeur, A Biswas, D Jacobs, W Kress, I Lopez, J Soares, Proceedings of the 12th European Conference on Computer Vision (ECCV), October 2012.
  - "Plant leaf recognition using shape features and colour histogram with k-nearest neighbor classifier," T Munisami, M Ramsurn, S Kishnah, S Pudaruth, Second International Symposium on Computer Vision and the the Internet, 2015.
  - "Leaves recognition system using a neural network," B Sekeroglu, Y Inan, 12th International Conference on Application of Fuzzy Systems and Soft Computing, August 2016.
  - "Deep Residual Learning for Image Recognition," Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015.