Deep Learning for Plant Identification

Ivica Dimitrovski

MAESTRA
LEARNING FROM MASSIVE, INCOMPLETELY ANNOTATED, AND STRUCTURED DATA

UKIM, Skopje

Outline

- Introduction
- Dataset
- Overview of Plant identification approaches
- Plant identification using deep learning
- Experiments and Results
- Issues and Problems
- Conclusions and Further work

Introduction

- Identifying plant species is usually impossible for the general public, and often a difficult task for professionals, such as farmers or wood exploiters and even for the botanists themselves
- Image-based plant identification is the most promising solution towards bridging the botanical taxonomic gap
- Mobile applications: LeafSnap and Pl@ntNet

Introduction

- In 2015, 2,328,502 millions queries have been submitted by the users of the Pl@ntNet mobile apps but only less than 3% of them were finally shared and collaboratively validated [*Plant identification in an open-world, Goeau et al., 2016*]
- It is crucial to boost the performance of the machine learning algorithms for automated identification
- Collecting and building appropriate training data is becoming one of the most central problem for solving the taxonomic gap problem [*LifeCLEF Plant Identification Task 2015, Goeau et al., 2015*]

Dataset

- Pl@ntView dataset containing 113,205 images

- Two versions of the Pl@ntView dataset
  - PlantCLEF 2014/2015 - the images from the test dataset are from known classes
  - PlantCLEF 2016 - open-world recognition problem

Pl@ntView dataset

Leaf diversity

Leaflets number variability on *Fraxinus angustifolia* Vahl (Narrow-leafed Ash)

Shooting conditions and used devices, *Acer platanoides L.* (Norway mapple)

Growing stage: two compound leaves **from the same tree !** *Gleditsia triacanthos* L. (Honey Locust)

Lobe number and deep of leaf lobes on *Ficus carica L.* (Common fig)

Spring 2012 Summer 2010

Autumn 2011 Winter 2011

Leaf at different growing stage of *Platanus x hispanica* Mill ex. Münchh. (London plane)

Autumnal variability of the lamina color on *Cotinus coggygria* Scop. (Eurasian smoketree)

Pl@ntView dataset

Flower diversity

COLOR

Brown White Green Rose Blue Yellow

Sym ¦ metry

Radial Bilateral

Structure

4 5 6 >>6

Number of petals

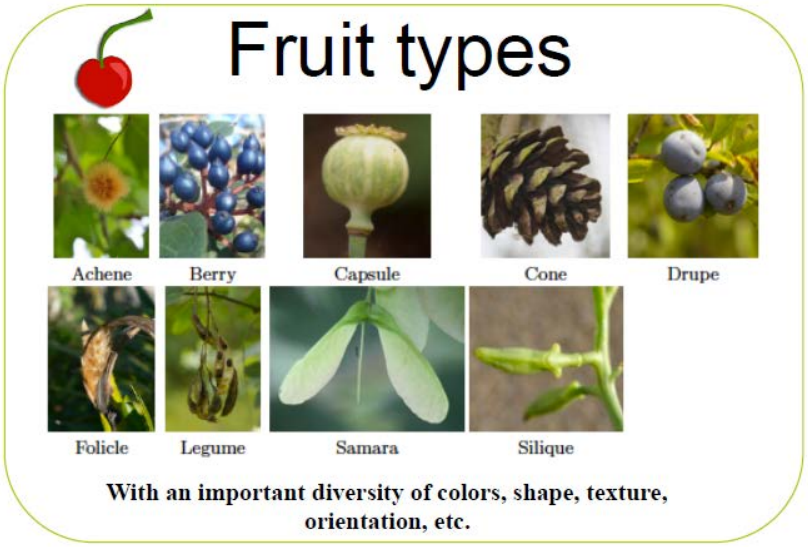
Orientation

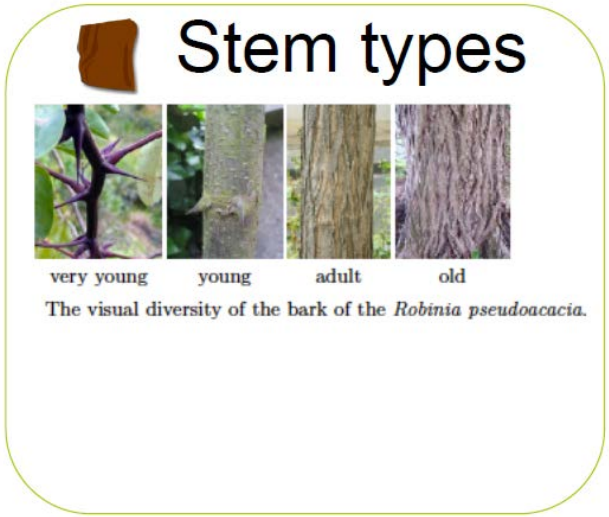
Face Profil

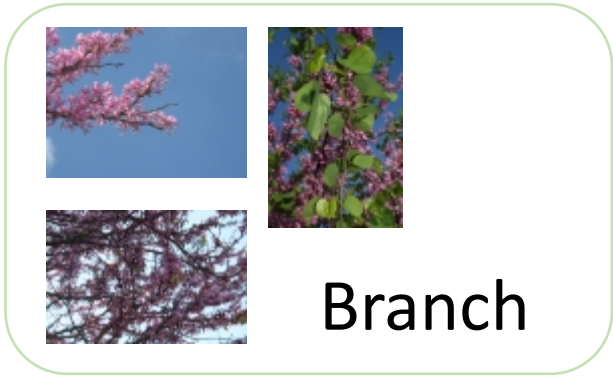
Size

small middle big

Pl@ntView dataset





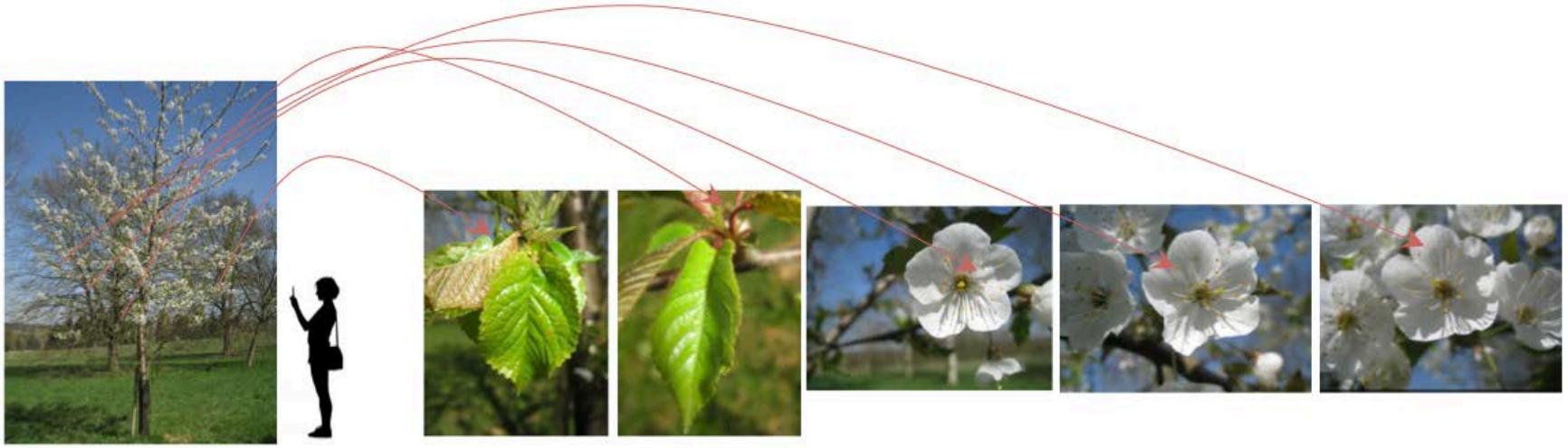


Meta-data for each image

- **ObservationId**: the plant observation ID from which several pictures can be associated
- **FileName**
- **MediaId**
- View **Content**: Branch, Entire, Flower, Fruit, Leaf, LeafScan, Stem
- **ClassId**: the class number ID that must be used as ground-truth. It is a numerical taxonomical number used by Tela Botanica
- **Species** the species names (containing 3 parts: the Genus name, the Species name, the author(s) who discovered or revised the name of the species)
- **Genus**: the name of the Genus, one level above the Species in the taxonomical hierarchy used by Tela Botanica
- **Family**: the name of the Family, two levels above the Species in the taxonomical hierarchy used by Tela Botanica
- **Date**: (if available) the date when the plant was observed,
- **Vote**: the (round up) average of the user ratings on image quality
- **Locality**: (if available) locality name, most of the time a town
- **Latitude & Longitude**: (if available) the GPS coordinates of the observation in the EXIF metadata, or, if no GPS information were found in the EXIF, the GPS coordinates of the locality where the plant was observed (only for the towns of metropolitan France)
- **Author**: name of the author of the picture,

Observation & Images

Set of one to several images depicting the same individual plant, observed by the same person, the same day, with the same device



PlantCLEF 2015

- 7 plant organs (7 types of image content)
- 1000 plant classes (1000 plant species)
- 113,205 images for over 43,000 observations
- 1 observation -> 1 to 5 images of a plant

MULTI QUERY CHALLENGE

(take in account all images when classifying an observation)

PlantCLEF 2016

- The test dataset is composed of unknown plant species and non-plant objects to create an open world plant dataset
  - recognizing the known plant species
  - but also rejecting unknown plants and objects

PlantCLEF 2016

- Open-set or open-world recognition is a challenging task in computer vision where samples from unknown classes could be presented
  - robust to unknown and never seen categories
- Beyond the brute-force classification across the known classes of the training set, a big challenge is thus to automatically reject the false positive classification hits that are caused by the unknown classes

PlantCLEF 2016

- The test set is composed of 8000 images
  - 4633 labeled with one of the 1000 known classes of the training set
  - 3367 labeled as new unknown classes
  - among the 4633 images of known species, 366 were tagged as invasive according to a selected list of 26 potentially invasive species

Overview of Pant Identification approaches

- Image segmentation
- Feature extraction
- Classification
- Fusion

Image segmentation

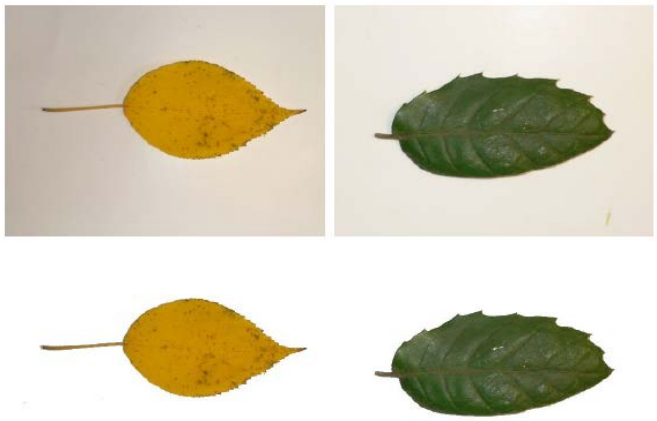
- Segmentation of the main objects and extracting the ROI often results in removing the irrelevant background which could introduce noise to a supervised classifier
- Different segmentation approaches for the different plant organs [IBM Research Australia at LifeCLEF2014: Plant Identification Task, Chen et al. 2014]

Flower and fruit

- Assumption that a flower or fruit is usually “more red” compared to leaves
- Locate the regions of which their red channel is larger than the green channel
- Each color image is converted to a gray-value image, and the localized “red” regions are then employed as initial masks in segmenting the flower or fruit in the gray-value image using a active contour method
- Compute the minimum bounding box of the flower or fruit mask as the ROI

Leaf Scan

- Leaf with a light background with variations among the background colors
- Normalization of the background with a consistent white color
- Convert the color image into a gray-value image and apply Otsu method to compute a threshold
- The pixels in the gray-value image that are smaller than the threshold are labeled as background, and all background pixels in the corresponding color image are assigned with a white color



Leaf

- The leaf is usually located in the center area of the picture and there are typically some margin between the leaf boundary and the picture border
- Extract the object that is located in the center of the picture

Stem

- The stem in usually located in the center of the image
- Convert the color image to a gray-value image and create a central mask on the image by cropping %25 from left and %25 from right
- Apply active contour on the gray-value image using the mask
- The bounding box of the resulting area is the ROI for the stem image

Feature extraction

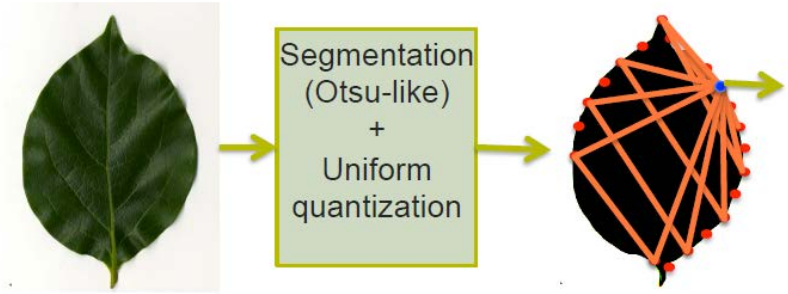
- Local features extraction around interest point in the segmented images
  - rotation invariant Local Binary Pattern (ri-LBP); SIFT; SURF; Fourier histogram; Edge Orientation Histogram; RGB histogram, LUV histogram; HSV histogram
- Visual index of local features (Pl@ntNet's participation at LifeCLEF 2014 Plant Identification Task, Goeau et al., 2014)
  - Local features form the train images are computed, compressed and indexed using Random Maximum Margin Hashing (RMMH) method to obtain unique hash tables
  - Each local feature from the query image is compressed with RMMH and its approximate k-nearest neighbors are searched by probing multiple neighboring buckets in the consulted hash table

Feature extraction

- Bag of Visual Words
  - Visual codebook for each plant organ
  - Approximate k-means or Predictive Clustering Trees (PCTs)
    - 7M – 10M local descriptors
    - 200K – 400K clusters / visual words
  - TF-IDF
  - L2 distance
- Fisher Kernel feature coding
  - GMMs in SIFT space

Feature extraction

- Multiscale triangular shape descriptors
  - Local matching of shapes
  - Robust & fast to compute
  - Validated on various leaf databases



400 local features described with 30 triangles

- The shape boundary is represented by a sequence of N sample points uniformly distributed over the contour and numbered in a clockwise order
- Each contour point is represented by triangles computed at different scales

Classification

- K-NN classifier for each plant organ
- Support Vector Machines
- Different similarity metrics
- Different fusion schemes to obtain the final predictions [PI@ntNet's participation at LifeCLEF 2014 Plant Identification Task, Goeau et al., 2014]

Summary of the approaches

- Fisher Vector encoding for state-of-the-art results
  - extract dense SIFT and Color Moments (CMs) in the images
  - reduce each local feature to a 64-dim after using PCA
  - estimate GMM model with 512 components for producing Fisher Vector representations for the images
  - linear trained SVMs classifiers, one for each type of features

IBM Research Australia at LifeCLEF2014: Plant Identification Task, Chen et al. 2014

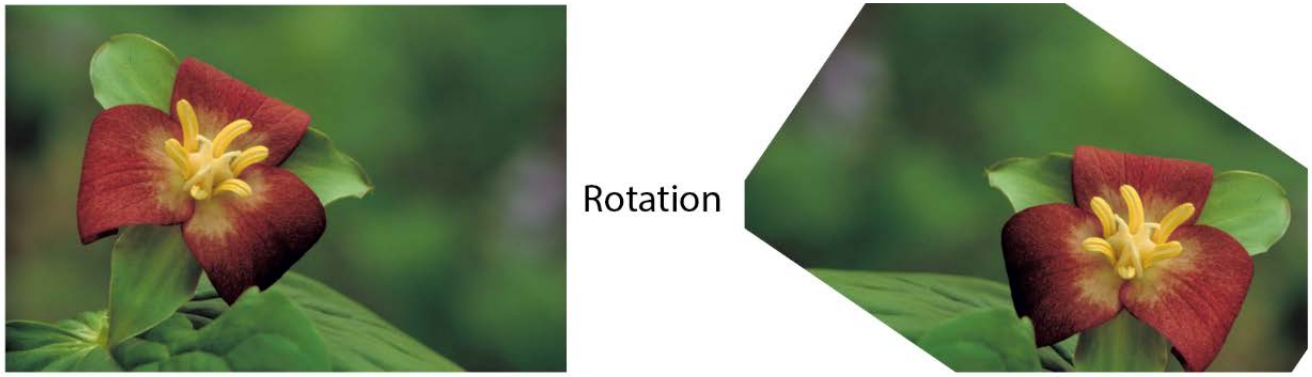
Deep learning approach

- Raw data is fed into the convolutional neural networks in multiple levels
  - Automatically discover high-level features for plant species recognition
- High computational complexity for training from scratch
- Pre-trained deep networks are fine-tuned for plant identification purposes
  - For example: VGGNet or GoogLeNet

Data augmentation

- Data augmentation is applied to decrease the chance of overfitting during training and to improve performance while testing
  - Increase the dataset size
- Artificial expansion of the dataset with data augmentation:
  - Augmentation by rotation
  - Augmentation by mirroring
  - Augmentation by skewing
- The white space after augmentation is filled with the dataset mean value

Augmentation by rotation



Original Image

Augmented Image

The white space after augmentation is filled with the dataset mean value.

Augmentation by mirroring

Original Image

H-FLIP

Augmented Image

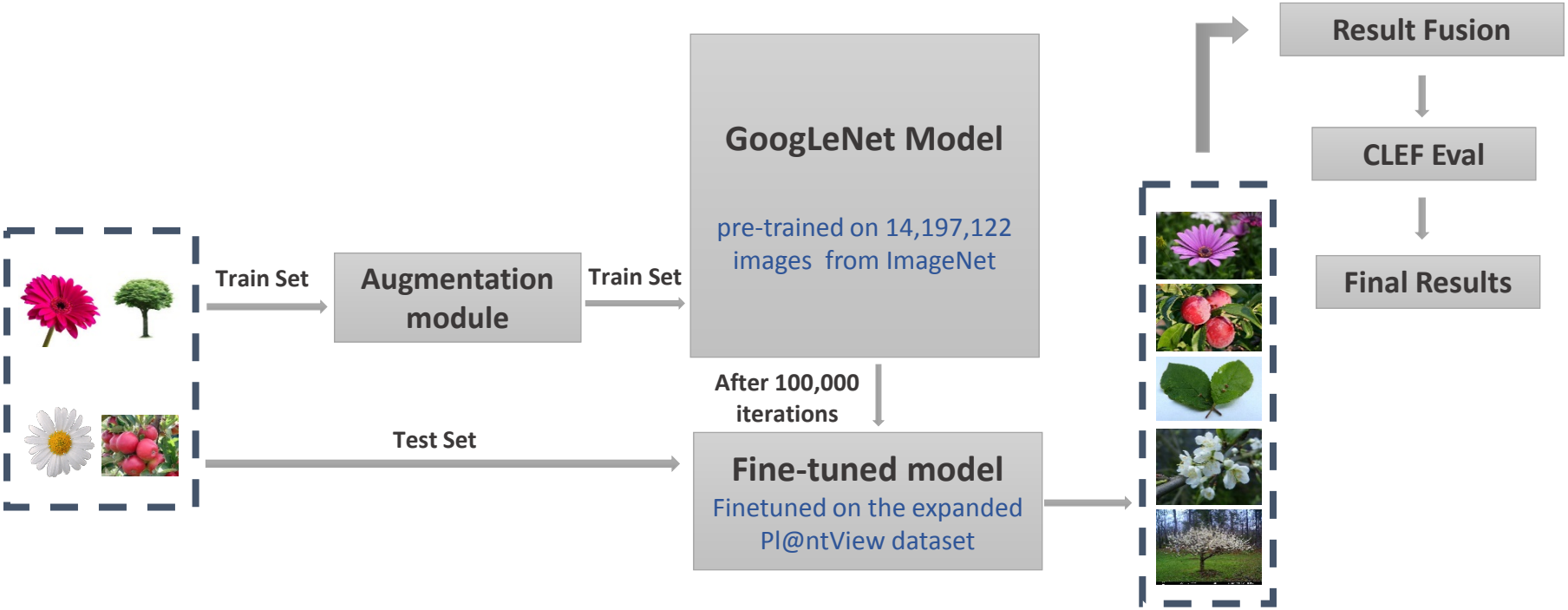
Augmentation by skewing

Original Image

Augmented Image

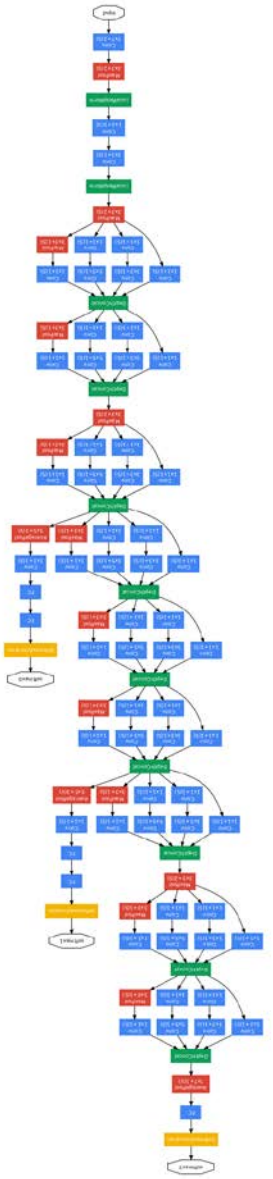
The white space after augmentation is filled with the dataset mean value.

System architecture



GoogLeNet Model

- Pretrained on the ImageNet dataset
- Implemented in Caffe
- Convolutional Neural Network with 22 (27) layers
- 224x224 receptive field
- RGB color space
- Inception architecture



Inception architecture

- Inception modules in convolutional networks are designed to allow for deeper and larger convolutional layers while at the same time allowing for more efficient computation
- This is done by using 1×1 convolutions with small feature map size:
  - for example, 192 28×28 sized feature maps can be reduced to 64 28×28 feature maps through 64 1×1 convolutions
  - Because of the reduced size, these 1×1 convolutions can be followed up with larger convolutions of size 3×3 and 5×5
  - In addition to 1×1 convolution, max pooling may also be used to reduce dimensionality
- In the output of an inception module, all the large convolutions are concatenated into a big feature map which is then fed into the next layer (or inception module)

GoogLeNet Layers

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Experiments

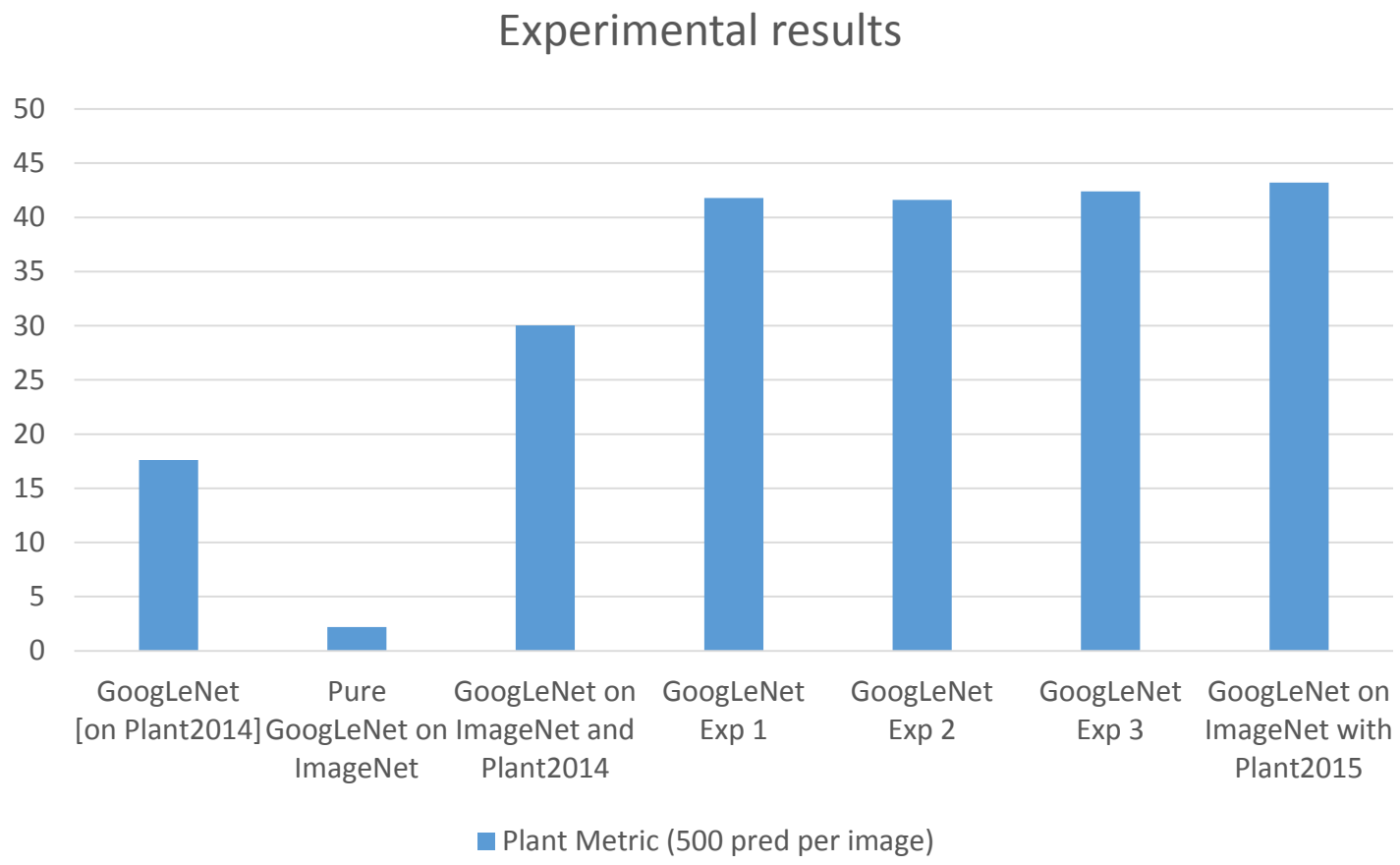
- The dataset size and diversity impact on the classification performance

- Data augmentation impact on the performance

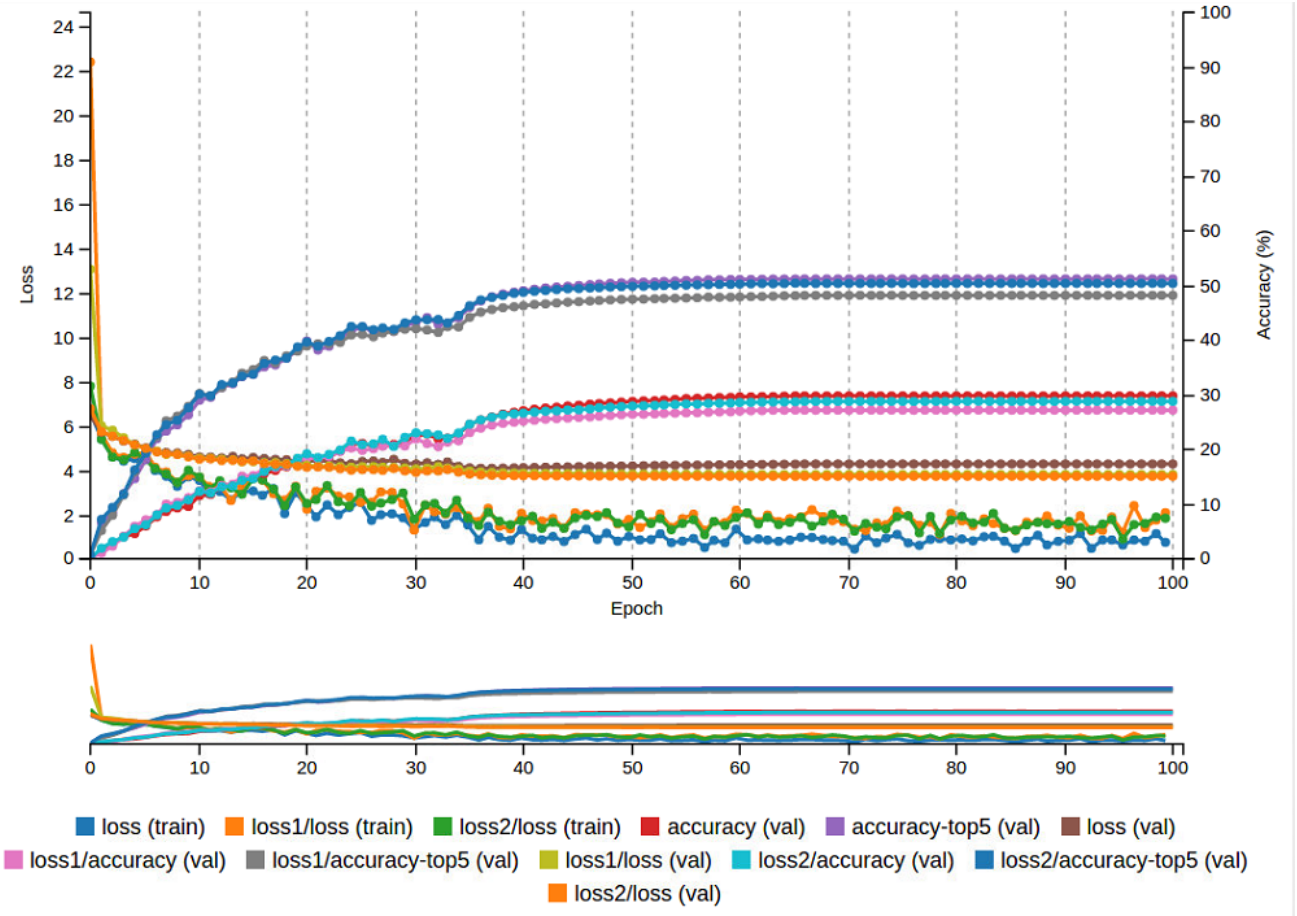
Experiments

Experiment	Dataset				
	Train	Validation	Test	Classes	Total
GoogLeNet [on Plant2014]	38252	9563	13146	500	60961
Pure GoogLeNet on Imagenet	14197122	9563	13146	500	14219831
GoogLeNet on ImageNet and Plant2014	38252	9563	13146	500	60961
GoogLeNet on ImageNet with Plant2014 Exp 1	76504	19126	13146	500	108776
GoogLeNet on ImageNet with Plant2014 Exp 2	114756	28689	13146	500	156591
GoogLeNet on ImageNet with Plant2014 Exp 3	153008	38252	13146	500	204406
GoogLeNet on ImageNet with Plant2015	73407	18352	13146	1000	104905

Results


Experimental results
50
45
40
35
30
25
20
15
10
5
0
GoogLeNet [on Plant2014]
Pure GoogLeNet on ImageNet
GoogLeNet on ImageNet and Plant2014
GoogLeNet Exp 1
GoogLeNet Exp 2
GoogLeNet Exp 3
GoogLeNet on ImageNet with Plant2015
Plant Metric (500 pred per image)

Learning curve



Irrelevant Sample Rejection for PlantCLEF 2016

- Separately fine-tune GoogLeNet model for a binary classification problem, i.e. plants vs. non-plants
- The training dataset contains plant images from LifeCLEF 2015 and non-plant images from ILSVRC competition
- Using the results from the binary classifier reject the images that are classified as non-plant and assign a low confidence score in the main plant identification system

Irrelevant Sample Rejection for PlantCLEF 2016

- The degree of novelty in the test set has a strong influence on the performance
- For example: In this year PlantCLEF 2016 when 25% of the queries still belong to a known class, none of the evaluated systems reach a mean average precision greater than 0.45 (to be compared to 0.83 in a closed world).

Additional problems

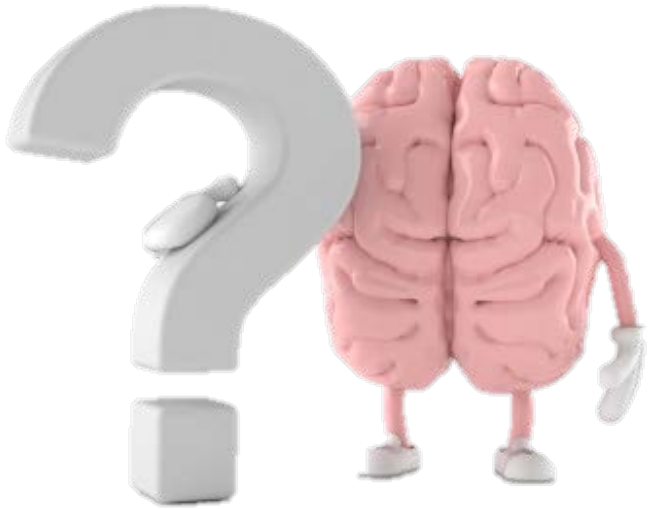
- The original GoogLeNet model requires input image of spatial size 224 × 224
- The images in the PlantCLEF 2014/2015 and 2016 dataset are arbitrarily sized
- Following the spatial size restriction in GoogLeNet model the input images has to be either cropped or warped which can lead to information loss

Additional problems

- Spatial Pyramid Pooling (SPP) layer
- The feature map is spatially divided into 1 × 1, 2 × 2, 4 × 4 = total of 21 regions
- Each region is then average pooled, producing a vector of fixed size 21 × 512 = 10752 (considering that the last convolution layer produces feature map of 512 channels)
- This conversion of arbitrarily sized feature map into a fixed size vector will allow the model to accept input image of any size

Conclusions and Further work

- CNNs pretrained on large dataset and fined tuned on augmented data to the specific domain to get best results
- CNNs is strongly affected by higher rates of images belonging to unknown classes
- Train different/multiple CNN networks (Ensemble of CNNs) and then combine the predictions to increase the classification performance
- Combination of hand crafted features in CNNs
- Use the hierarchy of taxonomy in the CNNs



Q&A

MAESTRA

LEARNING FROM MASSIVE, INCOMPLETELY ANNOTATED, AND STRUCTURED DATA