Bluefield (KDE TUT) at LifeCLEF 2016 Plant Identification Task

Siang Thye Hang
Atsushi Tatsuma
Masaki Aono
• Introduction and related work

• Modifying the VGG-16 layers CNN model
  – Spatial Pyramid Pooling (SPP)
  – Parametric Rectified Linear Unit (PReLU)

• Post processing
  – Unseen Category Sample Detection algorithm
  – Observation based identification

• Evaluation
  – Data augmentation and model training
  – Results and analysis

• Conclusion
Introduction (1/2)

- Accurate knowledge of plants is essential in agricultural development and biodiversity conservation.
- However, identifying a species can be difficult even for professionals.
- This leads to the consideration of using image retrieval technologies.
  - Convolutional Neural Network (CNN) is widely used in various image retrieval tasks.

query image

Daphne cneorum?
Rhododendron ferrugineum?
Oxalis articulata?
Silene acaulis?

...
Introduction (2/2)

- **Crowd sourced data is a cost efficient** method to provide data for image retrieval systems.
- However, they usually contain **higher level of noise and unknown objects**.
- In real world, instead of classification to a **fixed number of categories**, a good system is expected to be **robust to unseen categories**.

![Crowd sourced images (with noise)]

![Database with seen categories](Diospyros kaki, Prunus laurocerasus, Cyclamen hederifolium)

![Database with unseen categories](house plant Coleus, non living object, Garden plant Rosa)
Related work

• PlantCLEF 2014’s top performer: IBM Research Australia (IBM AU)
  – trained AlexNet CNN model
  – Fisher Vector encoded dense SIFT and Color Moments
  – Region of Interest based cropping and background removal

• PlantCLEF 2015’s top performer: Seoul National University Medinfo (SNUMED INFO)
  – Transfer learning on 5 GoogLeNet CNN Models, outputs combined

• Our objective is to design a system that is robust to unseen categories
  – Automatic, minimizing the need of human intervention
  – Without relying on transfer learning
  – Without external dataset
Proposed framework (train)

- Train image (augmented)
- Train label
- Modified VGG 16-layers CNN model
- Train
Proposed framework (train)

- **train image**
  - train image
  - modified CNN model (trained)
  - extract class scores (before softmax normalization)

- **train label**
  - train label
  - correctly predicted?
    - yes?
      - compute class thresholds
    - no?
      - ignore

- Decisions:
  - yes?
  - no?
Proposed framework (test)

- **Test image**
  - Modified CNN model (trained)
  - Extract class scores (before softmax normalization)
  - Compare

- **Test ObservationId**
  - Lower than threshold for all classes?
  - Yes?
    - Reject!
  - No?
    - Combine scores of same ObservationId
    - Combine class scores

- **Class thresholds (computed)**
  - 1: 10.8541
  - 2: 10.7569
  - ...
  - 1000: 12.5818

- **Final class scores (normalized)**

- **Softmax normalization**
Unseen Category Sample
Detection algorithm

Proposed framework

class scores (before softmax normalization)

modified CNN model (trained)

extract

compare

lower than threshold for all classes?

yes?

Reject!

no?

combine scores of same ObservationId

combine class scores

softmax normalization

Final class scores (normalized)
Proposed framework

Observation based identification

- Test image
- Test ObservationId
- Combined class scores
- No?
- Lower than threshold for all classes?
  - Yes? Reject!
  - No?
    - Combine scores of same ObservationId
    - No?
      - Extract class scores (before softmax normalization)
      - Compare
      - Yes?
        - Reject!
      - No?
        - Combined class scores
        - Softmax normalization
        - Final class scores (normalized)
Proposed framework

Modified VGG 16-layers CNN model

Extract class scores (before softmax normalization)

Compare lower than threshold for all classes?

Yes?

Reject!

No?

Combine scores of same ObservationId

Combined class scores

Softmax normalization

Final class scores (normalized)
The VGG-16 layers CNN model*

<table>
<thead>
<tr>
<th>Layer Description</th>
<th>Number of Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input (224×224 RGB image)</td>
<td></td>
</tr>
<tr>
<td>conv3-64, conv3-64</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-128, conv3-128</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-256, conv3-256, conv3-256</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-512, conv3-512, conv3-512</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-512, conv3-512, conv3-512</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>FC-4096, FC-4096</td>
<td></td>
</tr>
<tr>
<td>FC1000, softmax</td>
<td></td>
</tr>
</tbody>
</table>

The VGG-16 layers CNN model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>(224×224 RGB image)</td>
</tr>
<tr>
<td><strong>conv3-64, conv3-64</strong></td>
<td></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-128, conv3-128</strong></td>
<td></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-256, conv3-256, conv3-256</strong></td>
<td></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-512, conv3-512, conv3-512</strong></td>
<td></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-512, conv3-512, conv3-512</strong></td>
<td></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FC-4096, FC-4096</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FC1000, softmax</strong></td>
<td></td>
</tr>
</tbody>
</table>

The model consists of 16 convolutional layers followed by fully connected layers and a softmax layer for classification. The input is a 224×224 RGB image, and the model processes it through a series of convolution and pooling operations to extract features, which are then used for classification.
Modifying VGG 16-layers model

input (224×224 RGB image)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv3-64, conv3-64</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-128, conv3-128</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-256, conv3-256, conv3-256</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-512, conv3-512, conv3-512</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>conv3-512, conv3-512, conv3-512</td>
<td></td>
</tr>
<tr>
<td>maxpool</td>
<td></td>
</tr>
<tr>
<td>FC-4096, FC-4096</td>
<td></td>
</tr>
<tr>
<td>FC1000, softmax</td>
<td></td>
</tr>
</tbody>
</table>
Modifying VGG 16-layers model

input (arbitrarily sized RGB image)

- conv3-64, conv3-64
- maxpool
- conv3-128, conv3-128
- maxpool
- conv3-256, conv3-256, conv3-256
- maxpool
- conv3-512, conv3-512, conv3-512
- maxpool
- conv3-512, conv3-512, conv3-512
- Spatial Pyramid Pooling*
- FC-4096, FC-4096
- FC1000, softmax

convolutional layers

output of the last convolutional layer (arbitrarily sized)

fixed length representation (21×512-d)

fully connected (FC) layers

Modifying VGG 16-layers model

Rectified Linear Unit (ReLU)

\[ f(y) = y \]

\[ f(y) = 0 \]

Parametric ReLU*

\[ f(y) = \alpha y \]


coefficient \( \alpha \) for the negative part is not constant and is adaptively learned
Proposed framework

Unseen Category Sample Detection algorithm
Unseen Category Sample Detection algorithm

- The output of the last fully connected layer (FC-1000) can be assumed as output score of 1000 binary classifiers.

- A sample will be classified to the class with highest score (One-versus-Rest).

we may safely infer that this sample **belongs to class C**
Unseen Category Sample Detection algorithm

- The output of the last fully connected layer (FC-1000) can be assumed as output score of 1000 binary classifiers.

- A sample will be classified to the class with highest score (One-versus-Rest).

We may infer that this sample belongs to either class A or B or C.
Unseen Category Sample Detection algorithm

- The output of the last fully connected layer (FC-1000) can be assumed as output score of 1000 binary classifiers.

- A sample will be classified to the class with highest score (One-versus-Rest).

We may infer that this sample does not belong to any class (of unseen category).

How “low” should we define the thresholds?
Unseen Category Sample Detection algorithm

We propose to define the thresholds (for each class) as “minimum score of correctly predicted training samples”

1. Assume scores $S_{M \times N}$ of $N$ classes and $M$ training samples are extracted from the last fully connected layer (FC-1000)
   - Incorrectly predicted samples are omitted, leaving $M' (\leq M)$ samples
   - Only correctly predicted scores $S'_{M' \times N}$ are used

2. Thresholds $t_1, \ldots, t_N$ are computed with class-wise minima

   $$t_i = \min_{k \in M'} S_{k,i}$$

3. During testing, any sample with scores lower than the computed thresholds for all of the classes will be rejected
Unseen Category Sample Detection algorithm

- When training a CNN model, a subset of samples are separated (from the training samples) for validation purpose.

- So far, thresholds $t_1, \ldots, t_N$ are computed based on training samples only.
  - How about validation samples?

- Similarly, thresholds based on validation samples $v_1, \ldots, v_N$ are computed.

- Let $Q$ be average of $v_i/t_i$ for $i = 1, \ldots, N$, we update the thresholds by multiplying $t_1, \ldots, t_N$ with $Q$.

  $$t'_i = Q t_i$$

- As number of validation samples is usually less than training samples, we should expect that majority of $v_i$ will be higher than $t_i$.
  - Only $v_i$ that are lower than $t_i$ are considered into computation of $Q$. 
Proposed framework

Observation based identification

- test image
- modified VGG 16-layers CNN model
  - extract class scores (before softmax normalization)
  - compare class scores
  - lower than threshold for all classes?
    - yes? Reject!
    - no? combine scores of same ObservationId
- test ObservationId
  - combine class scores
  - no?
  - yes?
- softmax normalization

Final class scores (normalized)
Observation based identification

- Scores of samples with the same ObservationId are summed
  - Some samples are easily distinguishable while some are not
  - Samples with same ObservationId will have the same scores for all classes

ObservationId 16237 (5 samples)

ObservationId 21782 (1 sample)

ObservationId 37494 (35 samples)
• Introduction and related work

• Modifying the VGG-16 layers CNN model
  – Spatial Pyramid Pooling (SPP)
  – Parametric Rectified Linear Unit (PReLU)

• Post processing
  – Unseen Category Sample Detection algorithm
  – Observation based identification

• Evaluation
  – Data augmentation and model training
  – Results and analysis

• Conclusion
The PlantCLEF 2016 dataset

• 113204 labelled images
  – 111156 images to train CNN model
  – 2048 images to validate CNN model (every half epoch)

• 8000 unlabeled test images
Data Augmentation

• Train images
  – Resize such that shorter side becomes 224 and 336 (two scales) while preserving aspect ratio
  – Random cropping to 224×224
  – Random horizontal flipping

• Validation & test images
  – Resize such that shorter side becomes 224 while preserving aspect ratio
  – No cropping
  – No flipping
Training the modified CNN model

• Model is initialized with Xavier’s method*
• Learning rate 0.01 to 0.0001, batch size 50
  – 0.01 for 30 epochs
  – 0.001 for 15 epochs
  – 0.0001 for 8 epochs
• Final validation accuracy 0.626

Evaluation metric

- Mean Average Precision
  - including unknown classes and queries (official MAP)
  - restricted to a black list of potentially invasive species
  - ignoring unknown classes and queries
Submitted runs

Run1: Threshold $t$ based on train images
Run2: Threshold $t'$ based on train & validation images
Run3: Threshold $t$ based on train images, use ObservationId
Run4: Threshold $t'$ based on train & validation images, use ObservationId

<table>
<thead>
<tr>
<th>Run</th>
<th>Official score</th>
<th>MAP restricted to potentially invasive species</th>
<th>MAP ignoring unknown classes and queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.611</td>
<td>0.600</td>
<td>0.692</td>
</tr>
<tr>
<td>2</td>
<td>0.611</td>
<td>0.600</td>
<td>0.693</td>
</tr>
<tr>
<td>3</td>
<td>0.736</td>
<td>0.718</td>
<td>0.820</td>
</tr>
<tr>
<td>4</td>
<td>0.742</td>
<td>0.717</td>
<td>0.827</td>
</tr>
</tbody>
</table>
## Official Evaluation Results

<table>
<thead>
<tr>
<th>Run</th>
<th>Official score MAP</th>
<th>MAP restricted to potentially invasive species</th>
<th>MAP ignoring unknown classes and queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluefield Run 4</td>
<td>0.742</td>
<td>0.717</td>
<td>0.827 (with ObservationId)</td>
</tr>
<tr>
<td>SabanciUGebzeTU Run 1</td>
<td>0.738</td>
<td>0.704</td>
<td>0.806</td>
</tr>
<tr>
<td>SabanciUGebzeTU Run 3</td>
<td>0.737</td>
<td>0.703</td>
<td>0.807</td>
</tr>
<tr>
<td>Bluefield Run 3</td>
<td>0.736</td>
<td>0.718</td>
<td>0.820 (with ObservationId)</td>
</tr>
<tr>
<td>SabanciUGebzeTU Run 2</td>
<td>0.736</td>
<td>0.683</td>
<td>0.807</td>
</tr>
<tr>
<td>SabanciUGebzeTU Run 4</td>
<td>0.735</td>
<td>0.695</td>
<td>0.802</td>
</tr>
<tr>
<td>CMP Run 1</td>
<td>0.710</td>
<td>0.653</td>
<td>0.790</td>
</tr>
<tr>
<td>LIIR KUL Run 3</td>
<td>0.703</td>
<td>0.674</td>
<td>0.761</td>
</tr>
<tr>
<td>LIIR KUL Run 2</td>
<td>0.692</td>
<td>0.667</td>
<td>0.744</td>
</tr>
<tr>
<td>LIIR KUL Run 1</td>
<td>0.669</td>
<td>0.652</td>
<td>0.708</td>
</tr>
<tr>
<td>UM Run 4</td>
<td>0.669</td>
<td>0.598</td>
<td>0.742</td>
</tr>
<tr>
<td>CMP Run 2</td>
<td>0.644</td>
<td>0.564</td>
<td>0.729</td>
</tr>
<tr>
<td>CMP Run 3</td>
<td>0.639</td>
<td>0.590</td>
<td>0.723</td>
</tr>
<tr>
<td>QUT Run 3</td>
<td>0.629</td>
<td>0.610</td>
<td>0.696</td>
</tr>
<tr>
<td>Floristic Run 3</td>
<td>0.627</td>
<td>0.533</td>
<td>0.693</td>
</tr>
<tr>
<td>UM Run 1</td>
<td>0.627</td>
<td>0.537</td>
<td>0.700</td>
</tr>
<tr>
<td>Floristic Run 1</td>
<td>0.619</td>
<td>0.541</td>
<td>0.694</td>
</tr>
<tr>
<td>Bluefield Run 1</td>
<td>0.611</td>
<td>0.600</td>
<td>0.692 (without ObservationId)</td>
</tr>
<tr>
<td>Bluefield Run 2</td>
<td>0.611</td>
<td>0.600</td>
<td>0.693 (without ObservationId)</td>
</tr>
<tr>
<td>BME TMIT Run 4</td>
<td>0.174</td>
<td>0.144</td>
<td>0.213</td>
</tr>
</tbody>
</table>
Seen categories: correctly predicted
Seen categories: *incorrectly* predicted (1/2)

(100%) *Ailanthus altissima*

(0%) *Ruscus aculeatus* (ground truth)
Sevent categories: **incorrectly predicted (2/2)**

**Query image**

**Shares ObservationId with**

- **(43%)** *Diplotaxis erucoides*
- **(41%)** *Hesperis matronalis*
- **(11%)** *Cyclamen hederifolium* (ground truth)
Comparing rejection algorithm

- By computing the **rate of change** from “MAP ignoring unknown classes and queries” to “Official score MAP (including unknown classes and queries)”, we may find out **contribution of rejection algorithm**
  - Less rate of change may imply better rejection algorithm
  - Apparently our rejection algorithm is generally **weaker than other teams**

<table>
<thead>
<tr>
<th>Run</th>
<th>MAP ignoring unknown classes and queries (A)</th>
<th>Official score MAP (B)</th>
<th>Rate of change from A to B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluefield Run 4</td>
<td>0.827</td>
<td>0.742</td>
<td>-10.28%</td>
</tr>
<tr>
<td>SabanciU Gebze TU Run 1</td>
<td>0.806</td>
<td>0.738</td>
<td>-8.44%</td>
</tr>
<tr>
<td>CMP Run 1</td>
<td>0.790</td>
<td>0.710</td>
<td>-10.13%</td>
</tr>
<tr>
<td>LIIR KUL Run 3</td>
<td>0.761</td>
<td>0.703</td>
<td>-7.62%</td>
</tr>
<tr>
<td>UM Run 4</td>
<td>0.742</td>
<td>0.669</td>
<td>-9.84%</td>
</tr>
<tr>
<td>QUT Run 3</td>
<td>0.696</td>
<td>0.629</td>
<td>-9.63%</td>
</tr>
<tr>
<td>Floristic Run 3</td>
<td>0.693</td>
<td>0.627</td>
<td>-9.52%</td>
</tr>
<tr>
<td>BME TMIT Run 4</td>
<td>0.213</td>
<td>0.174</td>
<td>-18.31%</td>
</tr>
</tbody>
</table>
Result analysis: unseen categories

* Percentage are confidence (softmax normalized score) of first prediction

100% 100% 34% 60% 0%
57% 54% 93% 32% 27%
100% 100% 34% 60% 0%
Result analysis: unseen categories

around half of 8000 test samples are of unseen categories

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Number of samples rejected</th>
<th>Correctly rejected</th>
<th>Incorrectly rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$ (Run 1, 3)</td>
<td>195</td>
<td>139</td>
<td>56</td>
</tr>
<tr>
<td>$t'$ (Run 2, 4)</td>
<td>69</td>
<td>56</td>
<td>13</td>
</tr>
</tbody>
</table>
Conclusion

- We attempted to design a plant image identification system that is robust to unseen categories.

- Factors that contribute the most to our system:
  - Observation based identification (exploitation of ObservationId)
  - Multiscale (two scales) input images to train the CNN model
  - Arbitrarily sized input image during testing (Spatial Pyramid Pooling)

- To further improve our system:
  - We should train the CNN model longer ( > 100 epochs)
  - Randomly rotate input images
  - Randomly scale input images (instead of just two scales)
  - Exploit other metadata (GPS, Genus, Family, ...)

38