**Machine Learning on Melaleuca Herbarium Sheets**

Set up

* Two directories were set up each with 30 herbarium sheets inside – a directory with sheets that will be used to train the machine, the “train” directory, and another to test the machine’s detection, “test” directory. These sheets were randomly selected all within the Melaleuca genus.
  + This is the number of sheets we had set out to begin with. In actuality 32 sheets were used to train the machine.
  + Additional sheets: We checked for species bias to ensure at least 2 sheets of each species had been labelled
* Through anaconda, a graphic program “LabelMe” was used to label the herbarium sheets. Kentaro Wada. (2016). “*labelme: Image Polygonal Annotation with Python*”. <<https://github.com/wkentaro/labelme>>

Data set up

* All records lacking in their GPS location were removed

Eucalyptus only

* Records of Eucalyptus sp. were then assigned their respective subgenus according to the species in question. This was done through the use of Nicolle (2021), Euclid, and Brooker (2000).
* The number of distinct species in each subgenus of Eucalyptus was counted. If there was less than 10 distinct species, the subgenus would be classified as “small”, if there were more, it would be classified as “big”.

Brooker, M. I. H. (2000). A new classification of the genus Eucalyptus L'Hér. (Myrtaceae). Australian Systematic Botany 13(1) 79 - 148

Nicolle, D. (2021). Classification of the eucalypts (Angophora, Corymbia and Eucalyptus) Version 5. <http://www.dn.com.au/Classification-Of-The-Eucalypts.pdf>

Selecting sheets to label

* Using slice\_sample from the package tidyverse in R programming, one random sheet was selected for each species in ‘small’ subgenera of Eucalypts with no replacement
* For ‘big’ subgenera, Angophora sp. and Corymbia sp., ten random sheets in distinct species were selected with no replacement. These and the previous sheets was named as the ‘main lists’ for labelling.
* Back-up lists were also created for the above four main lists. This was done for ‘small’ subgenera of Eucalypts by filtering out sheets that were found in the ‘main lists’, then randomly selecting up to five different sheets for each species. For ‘big’ subgenera and other genera, sheets found in ‘main lists’ were filtered out and five random sheets were selected in each subgenera/genera.

Data dictionary

* A spreadsheet was created with the labels id, decimalLatitude, decimalLongitude, scientificName, Subgenus, Sampled, Reason, Leaf100, Leaf100B, Leaf90, Leaf50, Comment

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| id | The herbarium sheet ID | NSW:NSW:NSW 488173 |
| decimalLatitude | The decimal latitude of the location this record was collected | -26.33 |
| decimalLongitude | The decimal longitude of the location this record was collected | 150.67 |
| scientificName | The scientific name of this record | Eucalyptus tenuipes |
| Subgenus | Subgenus classification of the record (for Eucalyptus only) | Cuboidea |
| Sampled | Whether the record was sampled (Y/N) | Y |
| Reason | Left blank if record was sampled, provide reason if record was not sampled | Not available |
| Leaf100 | Number of Leaf100 sampled | 3 |
| Leaf100B | Number of Leaf100B sampled | 0 |
| Leaf90 | Number of Leaf90 sampled | 1 |
| Leaf50 | Number of Leaf50 sampled | 4 |
| Comment | Any comments including limited number of leaves or uncertainties | Leaf with gall, no Leaf100 |

Labelling

Labelling the leaves

* Through anaconda, a graphic program “*LabelMe*” was used to label the sheets under three labels BB, Leaf100, Leaf100B and Leaf50 using the ‘Create Polygons’ function.
* As we were aiming for measurements on solely the leaf blades, the petioles were excluded. However, for Eucalypts it was difficult to define where one segment started and the other ended thus an approximation was used as can seen in Fig. 2 and Fig. 3.
* Bounding boxes were first drawn to include at least one Leaf100/Leaf100B when possible. All leaves with an area of greater than 50% were labelled with the labels below.

Labels used in *LabelMe*:

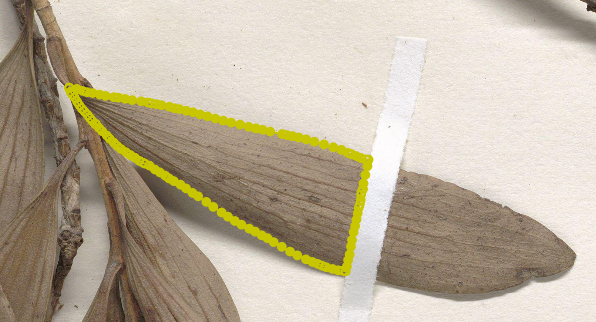
* **BB –** Bounding box. This was selected to contain at least one Leaf100/Leaf100B when possible. A suitable size was selected based on leaf size of specimen, with an average of 6 total labelled leaves per sheet.
* **Leaf100** – Complete leaves. No abnormal indentation that indicated herbivory, and no part of the leaf was covered by another. An example can be seen in Fig. 2 below.
* **Leaf100B** – Complete leaves, blemished. Minor abnormal indentation observed that indicate herbivory or cracks.
* **Leaf90** – Partial leaves. Leaves that had more than 90% of the blade visible, the remaining 10% may be from herbivory, coverage or bending of the leaf tip.
* **Leaf50** – Partial leaves. Leaves that had less than 90%, but more than 50%, of the blade visible. An example can be seen in Fig. 3 below

A picture containing graphical user interface

Description automatically generated

*Figure 2. Showing the point distance along a leaf, and examples of Leaf100 (Red), and LeafLess90 (Yellow)*

* It is to be noted that if the leaf was covered completely across by any object, the labelling would not go around that object. As shown in Fig. 3.



*Figure 3. Leaf50 showing how the labelling does not include external objects, which in this case is the tape*

## Documented abnormal circumstances

|  |  |  |
| --- | --- | --- |
| Description | Solution | Example |
| Two BB on one sheet due to Leaf100 being found on an isolated part of the leaf | Duplicate the image, with one bounding box on each sheet. Try to prevent this from happening, and only carry out if necessary. | NSW 307238 |
| Highly abnormal sized/Young leaves | Do not label unless Leaf50. This will teach the machine to ignore outliers like this. | NSW 355377 |
| Twisted leaf | Label the larger side of the leaf, do not follow the twist and label the whole leaf. Label appropriately as Leaf90 or Leaf50. | NSW 380818 |
| Deformed leaf; e.g., gall affecting leaf shape | Take note of the sheet and label as Leaf100B | NSW 488173 |

Other notes

* No interpolate across tape
* Include fragmented leaves in the bounding box, but do not label it. This will inform the machine to neglect them.

The code

Errors encountered:

Numpy ndarray incompatibility

* **Error Code**: ValueError: numpy.ndarray size changed, may indicate binary incompatibility. Expected 88 from C header, got 80 from PyObject
* **Reason**: Likely due to incompatibility in version with the Python and numpy.
* **Fix**: Uninstallation and reinstallation of numpy to the correct version.  
  pip uninstall numpy  
  pip install numpy==1.20

Creating an environment on user’s home directory through a .yml file

* **Reason**: Error with pytorch environment
* **Fix**: Uninstallation and reinstallation of pytorch followed with creation of the environment

conda remove -n pytorch

#Then, working on your home directory, copy over the yaml

cp /srv/scratch/cornwell/code/mc/environment.yml

#Setting up a gpu on katana  
qsub -I -l select=1:ncpus=8:ngpus=1:mem=46gb,walltime=2:00:00

#Creating the environment directory

mkdir /srv/scratch/*[home directory]/*envs

conda create --prefix /srv/scratch/*[home directory]*/envs/d2

conda activate /srv/scratch/*[home directory]*/envs/d2

module load cublas/10.1

module load cuda/10.1

module load cudnn/7.6.1-cuda10.1

module load nccl/2.7.8

conda env create -f environment.yml

Import errors due to lack of permission

* **Error Code**: Import Error: [Location]: cannot open shared object file: Permission denied
* **Reason**: The conda environment was located in a shared directory, with the files required used by the user who was not the owner of the files.
* **Fix**: Creating a new conda environment in the home directory of the owner, with first removing the initial pytorch environment if it is present.

#Uninstalling the initial pytorch   
conda uninstall pytorch

#Setting up a gpu on katana  
qsub -I -l select=1:ncpus=8:ngpus=1:mem=46gb,walltime=2:00:00

#Creating the environment directory

mkdir /srv/scratch/*[home directory]/*envs

conda create --prefix /srv/scratch/*[home directory]*/envs/d2

conda activate /srv/scratch/*[home directory]*/envs/d2

module load cublas/10.1

module load cuda/10.1

module load cudnn/7.6.1-cuda10.1

module load nccl/2.7.8

conda install pytorch=1.6 torchvision torchaudio cudatoolkit=10.1 -c pytorch

pip install opencv-python

pip install pyyaml==5.1

python -m pip install detectron2 -f <https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch1.6/index.html>

# Later, when running, activate as follows

# CONDA\_ENVS\_PATH=~miniconda3/envs:/srv/scratch/cornwell/code/envs

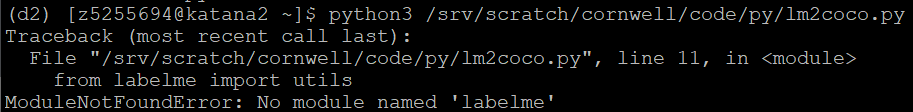
# conda activate /srv/scratch/cornwell/code/envs/d2

* **Running lm2coco.py on the server**
* Module Not Found Error – no module named ‘labelme’
* Attempt 1: (failed)  
  conda install labelme  
  python/srv…/code/py/lm2coco.py – output flower.json – classes Flower… -- polyORbb bb
* Attempt 2: Running it on local anaconda (failed)

Text

Description automatically generated

* Attempt 3: Running it on the homedrive python environment

****

* Attempt 4: Running it on a new conda environment

Install pip 🡪 label me 🡪 run lm2coco.py

conda create --n *envname*conda activate *labelenv*

conda install pip

pip install labelme

Text

Description automatically generated

Things to add to protocol: