Reproducibility review of: Benchmarking Invasive Alien Species Image Recognition Models for a Citizen Science Based Spatial Distribution Monitoring

Daniel Nüst

2022-06-10

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF at https://osf.io/k78eb/. To cite the report use

> Nüst, Daniel. 2022. Reproducibility Review of: Benchmarking Invasive Alien Species Image Recognition Models for a Citizen Science Based Spatial Distribution Monitoring. OSF. https://doi.org/10.17605/OSF.IO/K78EB

Reviewed paper

> *Niers, T., Stenkamp, J., Jakuschona, N. P., Bartoschek, T., and Schade, S.: Benchmarking Invasive Alien Species Image Recognition Models for a Citizen Science Based Spatial Distribution Monitoring, AGILE GIScience Ser., 3, 10, https://doi.org/10.5194/agile-giss-3-10-2022*

Summary

The article presents a comparison of seven image-based species recognition models, which were benchmarked against a set of species. Selected model executions were successfully reproduced as part of this reproducibility review, without any errors. The outputs were manually compared on a sample basis and match the result data shared privately by the authors; no summary statistics were recalculated. The authors provided the used data privately, but all code and good documentation is available online and properly deposited and cited using a data repository. Only two of the four online classification APIs were tested due to the requirement of registering accounts, therefore this reproduction is only partially complete.

Reproducibility reviewer notes

The authors provide a comprehensive Data and Software Availability section with materials published in several GitHub repositories but also deposited to Zenodo. Besides individual README files in each repository, the authors also reference an extensive technical report (https://doi.org/10.2760/97305). The authors also quickly provided access to the copyright protected image data that was used. The folder structure of the provided data is:

```
unique(dirname(list.files(here::here("017", "images_per_model"), recursive = TRUE)))
```

```
##  [1] "iNat2021_comp/candidates" "iNat2021_comp"
##  [3] "iNat2021_comp/general"    "iNaturalist/candidates"
##  [5] "iNaturalist/general"      "Microsoft/candidates"
##  [7] "Microsoft/general"        "NIA/candidates"
##  [9] "NIA/general"              "plantID/candidates"
## [11] "plantID/general"          "PlantNet/candidates"
## [13] "PlantNet/general"
```

So I start with retrieving the three projects and running the respective workflows.

Note: the steps below include updates made by the authors to include environment specifications and updated private image share. Please double check for the latest version of the Zenodo repositories. The authors also plan to add open example image data for testing to the repositories, which are not used in the following.

iNaturalist_Competition

```
git clone https://github.com/EibSReM/iNaturalist_Competition
cd iNaturalist_Competition

conda create -f environment.yml
# solving environment failed, mentions a number of ResolvePackageNotFound packages!
# create own env instead
conda create --name agile-017
conda activate agile-017

pip install torch torchvision
```

Download the data (5.6 GB) and unpacked it next to the repository. I set the data path to `../cvpr21_newt_pretrained_models/pt/inat2021_supervised_large_from_scratch.pth.tar`. I also downloaded the image data from a private share and unpacked it next to the repository. As instructed, I update the paths in `inference.py`, though it is not fully clear to me whether I should use the "candidates" or the "general" images. I continue with the "candidates", setting the path to `../images_per_model/iNaturalist/general/`.

```
python inference.py

# runs for about one minute, first few lines of the output shown below
```

```
['callosciurus_erythraeus_obs_02.png', 'lysichiton_americanus_obs_01.png', 'sciurus_carolinensis_obs_03.jpg',
 'gunnera_tinctoria_gold_03.jpg', 'oxyura_jamaicensis_obs_01.jpg', 'gunnera_tinctoria_gold_02.jpg',
 'threskiornis_aethiopicus_gold_02.jpg', 'gunnera_tinctoria_obs_02.png',
 # abbreviated [...]
, 'persicaria_perfoliata_gold_03.jpg', 'sciurus_carolinensis_gold_03.jpg']
this file is processed
callosciurus_erythraeus_obs_02.png
<class 'PIL.PngImagePlugin.PngImageFile'>
inference.py:59: UserWarning: volatile was removed and now has no effect. Use `with torch.no_grad():` instead.
  return torch.autograd.Variable(image, volatile=True)
{'label': 'Callosciurus erythraeus', 'probability': 0.6455265283584595}
{'label': 'Sciurus granatensis', 'probability': 0.056622788310050964}
{'label': 'Sciurus variegatoides', 'probability': 0.05055614560842514}
{'label': 'Alouatta palliata', 'probability': 0.04494091495871544}
{'label': 'Sciurus aureogaster', 'probability': 0.034268174320459366}
{'label': 'Callosciurus notatus', 'probability': 0.032729215919971466}
this file is processed
lysichiton_americanus_obs_01.png
<class 'PIL.PngImagePlugin.PngImageFile'>
{'label': 'Lysichiton americanus', 'probability': 0.15969304740428925}
```

```
{'label': 'Orontium aquaticum', 'probability': 0.12989430129528046}
{'label': 'Tulipa sylvestris', 'probability': 0.07191836833953857}
{'label': 'Phormium tenax', 'probability': 0.04441851004958153}
{'label': 'Heliconia psittacorum', 'probability': 0.042912933975458145}
{'label': 'Typha orientalis', 'probability': 0.018766984343528748}
this file is processed
sciurus_carolinensis_obs_03.jpg
<class 'PIL.JpegImagePlugin.JpegImageFile'>
{'label': 'Sciurus carolinensis', 'probability': 0.6847419142723083}
{'label': 'Sciurus aberti', 'probability': 0.05418068915605545}
{'label': 'Sciurus griseus', 'probability': 0.04411841928958893}
{'label': 'Macropus rufogriseus', 'probability': 0.03894677385687828}
{'label': 'Tamiasciurus douglasii', 'probability': 0.02306782826781273}
{'label': 'Sciurus aureogaster', 'probability': 0.015671176835894585}
this file is processed
gunnera_tinctoria_gold_03.jpg

# [...]
```

```
output <- read.table(file(here::here("017", "iNaturalist_Competition", "Output.txt")))
```

The created `Output.txt` file has 342 rows, the first few of which are

```
head(output)
```

```
##                                                                      V1
## 1 ../images_per_model/iNaturalist/general/callosciurus_erythraeus_obs_02.png
## 2   ../images_per_model/iNaturalist/general/lysichiton_americanus_obs_01.png
## 3    ../images_per_model/iNaturalist/general/sciurus_carolinensis_obs_03.jpg
## 4      ../images_per_model/iNaturalist/general/gunnera_tinctoria_gold_03.jpg
## 5      ../images_per_model/iNaturalist/general/oxyura_jamaicensis_obs_01.jpg
## 6      ../images_per_model/iNaturalist/general/gunnera_tinctoria_gold_02.jpg
##             V2           V3     V4        V5          V6     V7           V8
## 1 Callosciurus   erythraeus 0.6455   Sciurus granatensis 0.0566      Sciurus
## 2   Lysichiton   americanus 0.1597  Orontium   aquaticum 0.1299       Tulipa
## 3      Sciurus carolinensis 0.6847   Sciurus      aberti 0.0542      Sciurus
## 4      Gunnera    tinctoria 0.5073 Oplopanax    horridus 0.2140 Broussonetia
## 5       Oxyura  jamaicensis 0.7967 Bucephala     albeola 0.0529       Aythya
## 6      Gunnera    tinctoria 0.9651 Petasites    hybridus 0.0127     Wigandia
##              V9    V10       V11         V12    V13          V14         V15
## 1 variegatoides 0.0506  Alouatta    palliata 0.0449      Sciurus aureogaster
## 2    sylvestris 0.0719  Phormium       tenax 0.0444    Heliconia psittacorum
## 3       griseus 0.0441  Macropus rufogriseus 0.0389 Tamiasciurus   douglasii
## 4    papyrifera 0.0292 Dicksonia  antarctica 0.0285      Cyathea  medullaris
## 5       affinis 0.0405    Aythya      marila 0.0129         Anas    gracilis
## 6         urens 0.0084 Oplopanax    horridus 0.0051     Alocasia       odora
##      V16          V17         V18    V19
## 1 0.0343 Callosciurus     notatus 0.0327
## 2 0.0429        Typha  orientalis 0.0188
## 3 0.0231      Sciurus aureogaster 0.0157
## 4 0.0227      Corylus   americana 0.0206
## 5 0.0125         Anas  bahamensis 0.0093
## 6 0.0029    Heracleum  sosnowskyi 0.0021
```

A brief manual inspection confirms some overlap between a privately shared spreadsheet for the model data, only with some rounding errors. However, the number of rows does not match (authors table has 355). I do not know why, but the code seems to work.

TODO/HELP ME: I assume this data is summarised into the line "iNat2021" in Table 5, but it is unclear how these summaries were calculated.

MicrosoftSpeciesClassification

I work through the steps in the README. Since the provided Conda environment did not resolve, probably because of different OSes, I create a Python environment manually, following the instructions in an older version of the README:

```
# create and activate a new conda environment agile-017-microsoft
git clone https://github.com/EibSReM/MicrosoftSpeciesClassification
cd MicrosoftSpeciesClassification

conda install pytorch==1.2.0 torchvision==0.4.0 cudatoolkit=10.0 -c pytorch
pip install pretrainedmodels==0.7.4
pip install pillow==6.1.0
```

```
pip install progressbar2==3.51.0
pip install cupy-cuda100==7.3.0
pip install torchnet==0.0.4
pip install matplotlib pandas scikit-image
pip install pillow==6.1.0 # again, otherwise last command updates to 9.x
```

After adjusting the image path to `../images_per_model/Microsoft/candidates/` for my local setup, I run the script. I get an error message: `ImportError: libcuda.so.1: cannot open shared object file: No such file or directory` so I install CUDA development libraries with `sudo apt-get install nvidia-cuda-dev`.

```
python classify_images.py
```

It takes some time with my internet connection to download the classification and model data. The actual classification run only takes a few minutes. The final lines of output are shown below.

```
# abbreviated
Processing image 61 of 66
"Perca flavescens","0.045",
"Lutjanus griseus","0.033",
"Lontra canadensis","0.022",
Warning: latin name dasyatis americana not in lookup table
"Dasyatis americana","0.022",
"Thylogale","0.019",
Processing image 62 of 66
"Pistia stratiotes","0.821",
"Veratrum californicum","0.005",
"Frasera speciosa","0.005",
"Acanthus mollis","0.002",
"Helleborus foetidus","0.002",
Processing image 63 of 66
"Plestiodon obsoletus","0.037",
"Plestiodon skiltonianus","0.027",
"Plestiodon fasciatus","0.025",
"Plestiodon skiltonianus skiltonianus","0.020",
"Taricha granulosa","0.018",
Processing image 64 of 66
"Pycnonotus cafer","0.926",
"Pycnonotus jocosus","0.033",
"Pycnonotus barbatus","0.016",
"Anthornis melanura","0.000",
"Phainopepla nitens","0.000",
Processing image 65 of 66
"Celastrus orbiculatus","0.119",
"Aralia spinosa","0.080",
"Callicarpa americana","0.059",
"Toxicodendron radicans","0.039",
"Aralia nudicaulis","0.037",
Finished classifying 66 of 66 images (0 errors)
total runtime: 179.30823397636414
```

```
ms_output <- read.csv(file(here::here("017", "MicrosoftSpeciesClassification", "classification_output.csv")))
```

The created `classification_output.csv` file has 65 rows, the first few of which are

```
head(ms_output)
```

```
##       Pistia.stratiotes X0.927 Potamogeton.nodosus X0.001
## 1  Phytolacca americana  0.706 Verbesina virginica  0.036
## 2             Axis axis  0.717           Dama dama  0.032
## 3 Celastrus orbiculatus  0.680  Celastrus scandens  0.154
## 4  Phytolacca americana  0.780    Ricinus communis  0.028
## 5      Myocastor coypus  0.524  Erethizon dorsatum  0.296
## 6    Gambusia holbrooki  0.920    Gambusia affinis  0.052
##           Nelumbo.lutea X0.000       Veratrum.viride X0.000.1
## 1 Verbesina alternifolia  0.022 Verbesina encelioides    0.012
## 2 Odocoileus virginianus  0.018   Odocoileus hemionus    0.011
## 3       Orthilia secunda  0.023    Euonymus europaeus    0.008
## 4         Rivina humilis  0.006      Coriaria arborea    0.005
## 5      Castor canadensis  0.019      Hystrix cristata    0.011
## 6 Thalassoma bifasciatum  0.003     Carassius auratus    0.002
##             Veratrum.californicum X0.000.2
## 1                Persea americana    0.012
## 2 Odocoileus hemionus columbianus    0.008
## 3                Malus ×domestica    0.007
## 4              Prunus virginiana    0.004
## 5       Hydrochoerus hydrochaeris    0.010
```

```
## 6              Anguilla rostrata    0.002
##       ...images_per_model.Microsoft.candidates.pistia_stratiotes_obs_01.jpg
## 1  ../images_per_model/Microsoft/candidates/phytolacca_americana_gold_03.jpg
## 2             ../images_per_model/Microsoft/candidates/axis_axis_obs_01.jpg
## 3 ../images_per_model/Microsoft/candidates/celastrus_orbiculatus_gold_01.jpg
## 4   ../images_per_model/Microsoft/candidates/phytolacca_americana_obs_01.jpg
## 5      ../images_per_model/Microsoft/candidates/castor_canadensis_obs_03.jpg
## 6    ../images_per_model/Microsoft/candidates/gambusia_holbrooki_gold_01.jpg
```

Manually comparing the results for some images, e.g., `gambusia_holbrooki_obs_02.jpg` and `solenopsis_invicta_obs_02.jpg`, with the result table `Public_Models_Metadata` confirm the numbers match.

RequestCollectionComputerVisionAPIs

The authors do provide a Conda environment specification, but again, I assume due to different OSs, I install the required library manually:

```
# reuse conda environment agile-017
git clone https://github.com/EibSReM/RequestCollectionComputerVisionAPIs
cd RequestCollectionComputerVisionAPIs

pip install notebook
pip install requests

jupyter notebook apiRequests.ipynb
```

I register new accounts at *Plant.id* ("should receive API key within a few hours") and *Pl@ntNet* but skip *iNaturalist* and *NIA* because of the extra verification step via Email. Then I follow the instructions in the little form provided when running the last cell of the notebook.

Plant.id

Inputs:

- Directory with test images: `[.]/reviews-2022/reports/017/images_per_model/plantID/candidates/`
- Filename of result CSV: `plant.id-candidates.csv`
- API: `plantID`

Results are given within a minute or so; the data cannot be imported directly, as column numbers vary (possibly because multiple hits are given for each image):

```
plantid_output <- readLines(file(here::here("017", "RequestCollectionComputerVisionAPIs/plant.id-candidates.csv")))
plantid_output[1:4]
```

```
## [1] "pistia_stratiotes_obs_01.jpg,Pistia stratiotes,0.9902019571385497"
## [2] "phytolacca_americana_gold_03.jpg,Phytolacca americana,0.96652080290919,Phytolacca,0.011440228420710118"
## [3] "celastrus_orbiculatus_gold_01.jpg,Celastrus orbiculatus,0.951643142690765"
## [4] "phytolacca_americana_obs_01.jpg,Phytolacca americana,0.950753151274027,Phytolacca,0.013356158285156593"
```

PlantNet

Inputs:

- Directory with test images: `[.]/reviews-2022/reports/017/images_per_model/PlantNet/candidates/`
- Filename of result CSV: `plantnet-candidates.csv`
- API: `plantNet`

Example lines of cell output:

```
# [abbreviated]

200
{'language': 'en',
 'preferedReferential': 'the-plant-list',
 'query': {'images': ['7f1d704694a75857e63ebe763495d2f3'],
           'includeRelatedImages': False,
           'organs': ['auto'],
           'project': 'the-plant-list'},
 'remainingIdentificationRequests': 478,
 'results': [{'gbif': {'id': '3033896'},
              'score': 0.08309,
              'species': {'commonNames': ['Japanese barberry',
                                          "Thunberg's barberry",
                                          'Japanese berberis'],
                          'family': {'scientificName': 'Berberidaceae',
                                     'scientificNameAuthorship': '',
                                     'scientificNameWithoutAuthor': 'Berberidaceae'},
                          'genus': {'scientificName': 'Berberis',
                                    'scientificNameAuthorship': '',
                                    'scientificNameWithoutAuthor': 'Berberis'},
                          'scientificName': 'Berberis thunbergii DC.',
                          'scientificNameAuthorship': 'DC.',
                          'scientificNameWithoutAuthor': 'Berberis '
                                                         'thunbergii'}},
             {'gbif': {'id': '2889868'},
              'score': 0.02872,
              'species': {'commonNames': ['Acacia mistletoe',
                                          'Desert mistletoe',
                                          'Mesquite mistletoe'],
                          'family': {'scientificName': 'Santalaceae',
                                     'scientificNameAuthorship': '',
                                     'scientificNameWithoutAuthor': 'Santalaceae'},
                          'genus': {'scientificName': 'Phoradendron',
                                    'scientificNameAuthorship': '',
                                    'scientificNameWithoutAuthor': 'Phoradendron'},
                          'scientificName': 'Phoradendron californicum Nutt.',
                          'scientificNameAuthorship': 'Nutt.',
                          'scientificNameWithoutAuthor': 'Phoradendron '
                                                         'californicum'}},
             {'gbif': {'id': '9220780'},
              'score': 0.0282,
              'species': {'commonNames': ['Hawthorn',
                                          'Red hawthorn',
                                          'English hawthorn'],
                          'family': {'scientificName': 'Rosaceae',
                                     'scientificNameAuthorship': '',
                                     'scientificNameWithoutAuthor': 'Rosaceae'},
                          'genus': {'scientificName': 'Crataegus',
                                    'scientificNameAuthorship': '',
                                    'scientificNameWithoutAuthor': 'Crataegus'},
                          'scientificName': 'Crataegus monogyna Jacq.',
                          'scientificNameAuthorship': 'Jacq.',
                          'scientificNameWithoutAuthor': 'Crataegus monogyna'}},
             {'gbif': {'id': '3023221'},
              'score': 0.02658,
# [abbreviated]

200
{'language': 'en',
 'preferedReferential': 'the-plant-list',
 'query': {'images': ['e41b423e5d1ccb0f33d813433eaceae6'],
           'includeRelatedImages': False,
           'organs': ['auto'],
           'project': 'the-plant-list'},
 'remainingIdentificationRequests': 477,
 'results': [{'gbif': {'id': '2870583'},
              'score': 0.99731,
              'species': {'commonNames': ['Shellflower',
                                          'Water-cabbage',
                                          'Water-lettuce'],
                          'family': {'scientificName': 'Araceae',
                                     'scientificNameAuthorship': '',
                                     'scientificNameWithoutAuthor': 'Araceae'},
                          'genus': {'scientificName': 'Pistia',
                                    'scientificNameAuthorship': '',
                                    'scientificNameWithoutAuthor': 'Pistia'},
                          'scientificName': 'Pistia stratiotes L.',
                          'scientificNameAuthorship': 'L.',
                          'scientificNameWithoutAuthor': 'Pistia stratiotes'}}],
 'version': '2022-02-14 (5.1)'}

# [abbreviated]
```

After a few minutes, the output file is not updated anymore:

```
plantnet_output <- readLines(file(here::here("017", "RequestCollectionComputerVisionAPIs/plantnet-candidates.csv")))
plantnet_output[1:4]
```

```
## [1] "pistia_stratiotes_obs_01.jpg,Pistia stratiotes,0.99725"
## [2] "phytolacca_americana_gold_03.jpg,Phytolacca americana,0.97744,Phytolacca acinosa,0.00605,Phytolacca bogotensis,0.00223,Phytolacca escu
## [3] "celastrus_orbiculatus_gold_01.jpg,Celastrus orbiculatus,0.66886,Celastrus scandens,0.10653,Ilex verticillata,0.01817,Calycanthus flori
## [4] "phytolacca_americana_obs_01.jpg,Phytolacca americana,0.46878,Phytolacca bogotensis,0.06445,Phytolacca icosandra,0.04747,Pleuropetalum
```

Again, the manual comparison of images with the authors' result table confirms matching numbers with some rounding differences, e.g., for `pistia_stratiotes_obs_01.jpg`, but also some larger differences, e.g., for `celastrus_orbiculatus_gold_01.jpg`, see line three above and the original data as follows:

```
celastrus_orbiculatus_gold_01.jpg  Celastrus orbiculatus   0.77614 Celastrus scandens
0.13216 Celastrus paniculatus   0.00453 Arctium minus   0.00286 Callicarpa americana
0.00248 1   gc  b
```

This probably falls within model variability.

iNaturalist

This API was not tested for time constraints and the need for registration.

NIA

This API was not tested for time constraints and the need for registration.

Table 5

The summary statistics for Table 5 were not calculated as scripts/documentation on how to do that was missing.

Comments to the authors

- Good job overall, though having a single README that clear defines which part of your main Table 5 is based on which repo, or even a "parent repo" collecting all the information, would have been nice. I appreciate the transparency though about the different author contributions that is transparent through the different repositories.
- ~~When you have a step to download data, please do mention the download size.~~ README updated.
- ~~Include expected executing times in your instructions (experiences on your computer).~~ README updated.
- ~~Consider providing conda environment files so that versions are properly pinned and users only need to run one command to get a computational environment matching yours.~~ README updated.
- Follow your own instructions before submission to catch errors.
- ~~The account requests require detailed information about how many requests will be made - I don't know that, please provide that kind of detail when you ask users to register an account with a third party.~~ README updated.
- ~~It would be great if you could provide a handful of test images freely/openly with the expected result for demonstration purposes.~~ Repositories updated.
- Have a script for the calculation and aggregation of the statistics (*Table 5*), and publicly share your `Public_Models_Metadata` spreadsheet, ideally as a plain text file (CSV format); what could help with that: use the same structure for all output files.