2020 P. Borianne / CIRAD
Version 2.0 – February 2023

User guide

Checker was developed in the framework of the AgroDeep (AD) project; it allows to visually analyze, and if necessary to correct, the prediction errors of a specialized neural network. It is distributed under a Cecill-B license.

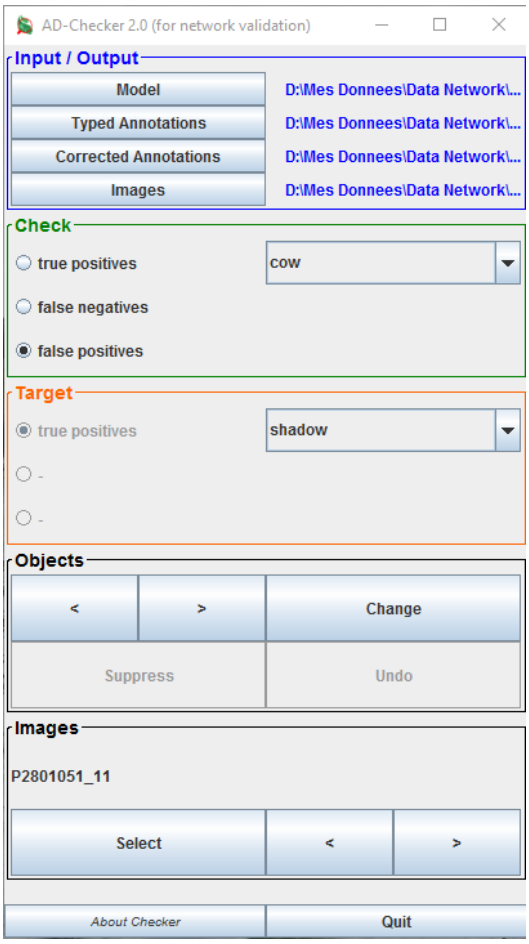


Figure 1 : IHM

Principle

The validation of a neural network consists, among other things, in comparing the expert's annotations to the network's predictions. This work leads to divide the objects into three distinct types:

- True Positives, made up of objects simultaneously annotated by the expert and detected by the network,
- False Positives, corresponding to objects detected by the network but not corresponding to any expert annotation,
- False negatives, corresponding to objects annotated by the expert but not detected by the network.

The result checking consists in making sure that the various objects present in the image are correctly "typed"; it is necessary to be able to correct errors that are not the responsibility of the network (for example FALSE FALSE POSITIVE which correspond to an expert annotation defect).

The Checker plugin allows you to quickly visualize the elements by type and if needed

1. to transfer an object from the False Positive type to the True Positive type, thereby correcting an expert annotation defect,
2. to delete an object of the type False Negatives, thereby correcting an excess annotation,
3. to reclassify an object, particularly in the case of multi-class networks

Input / Output block

Here are specified the data model to be used and the folders containing the images and type files respectively.

- The **Model** file to use: this is a TXT file that specifies the name and color of the object classes present in the images.

```
# model CowsAndDonkeys
# the color is defined by the rgb string where r, g and b are respectivelly the red,
green and blue componantes ;
# these components range from 0 to 255
#
MODELNAME   CowsAndDonkeys
OBJNAME     cow
OBJCOLOR    255 145 35
OBJNAME     donkey
OBJCOLOR    35 145 255
OBJNAME     shadow
OBJCOLOR    0 145 0
```

- The **Typed Annotations** folder: it contains the TXT files of the typed annotations coming from the validation processes of the AgroDeep platform; the network validation evaluates the pairing between expert annotations and network predictions: a **true positive** (TP) is a network prediction matched to an expert annotation, a **false positive** (FP) is an unpaired prediction and a **false negative** (FN) is an unpaired annotation. *False positives* are either annotation defects of the expert or detection or classification confusions of the network; *false negatives* are either detection defects of the network or annotation confusions of the expert.

	BX	BY	Width	Height	Name	Type
1	3586	2921	30	29	cow	TP
2	**3595**	**2926**	**24**	**27**	**cow**	**FP**
3	3365	2679	24	17	cow	TP
4	3158	2166	18	23	cow	TP
5	2788	2216	28	19	cow	TP
6	2827	2231	27	16	cow	TP
7	2882	2247	34	13	cow	TP
8	**2791**	**2215**	**26**	**18**	**cow**	**FN**
9	**2887**	**2242**	**27**	**18**	**cow**	**FP**
10	3191	1967	17	18	cow	TP
...	....					

- The **Corrected Annotations** folder: it contains the TXT files of annotations after taking into account the modifications made by the operator; these files can be used in the training processes of the AgroDeep platform.

ID	BX	BY	Width	Height	Name
1	3586	2921	30	29	cow
2	***3595***	***2926***	***24***	***27***	***shadow***
3	3365	2679	24	17	cow
4	3158	2166	18	23	cow
5	2788	2216	28	19	cow
6	2827	2231	27	16	cow
7	2882	2247	34	13	cow
~~*8*~~	~~*2791*~~	~~*2215*~~	~~*26*~~	~~*18*~~	~~*cow*~~
9	***2887***	***2242***	***27***	***18***	***cow***
10	3191	1967	17	18	cow
...	....				

- The **Images** folder: it contains Image files (png, jpg, JPEG,...) with the same radical as the files contained in the Annotations folder.

Check block

The filters used are specified here in terms of Type ("True Positive", "False Positive" or "False Negative") and class. Figure 1 specifies that the check will focus on false positives of the "cow" class..

Target block

This section indicates what change will be explicitly made by the operator; Figure 1 indicates that the current "false positive" of the "cow" class will be transformed into the "true positive" of the "shadow" class.

Objects block

Here are proposed the functionalities applicable to the objects:

- **<** : access to the previous object of the current image in the specified type,
- **>** : access to the next object of the current image in the specified type,
- **Change Type** : allows to modify the type or class of the current object,

- **Suppress** : removes the current object from the list of objects associated with the current image,
- **Undo** : cancels the last operation performed.

Functionalities are blocked / unblocked depending on the filters used.

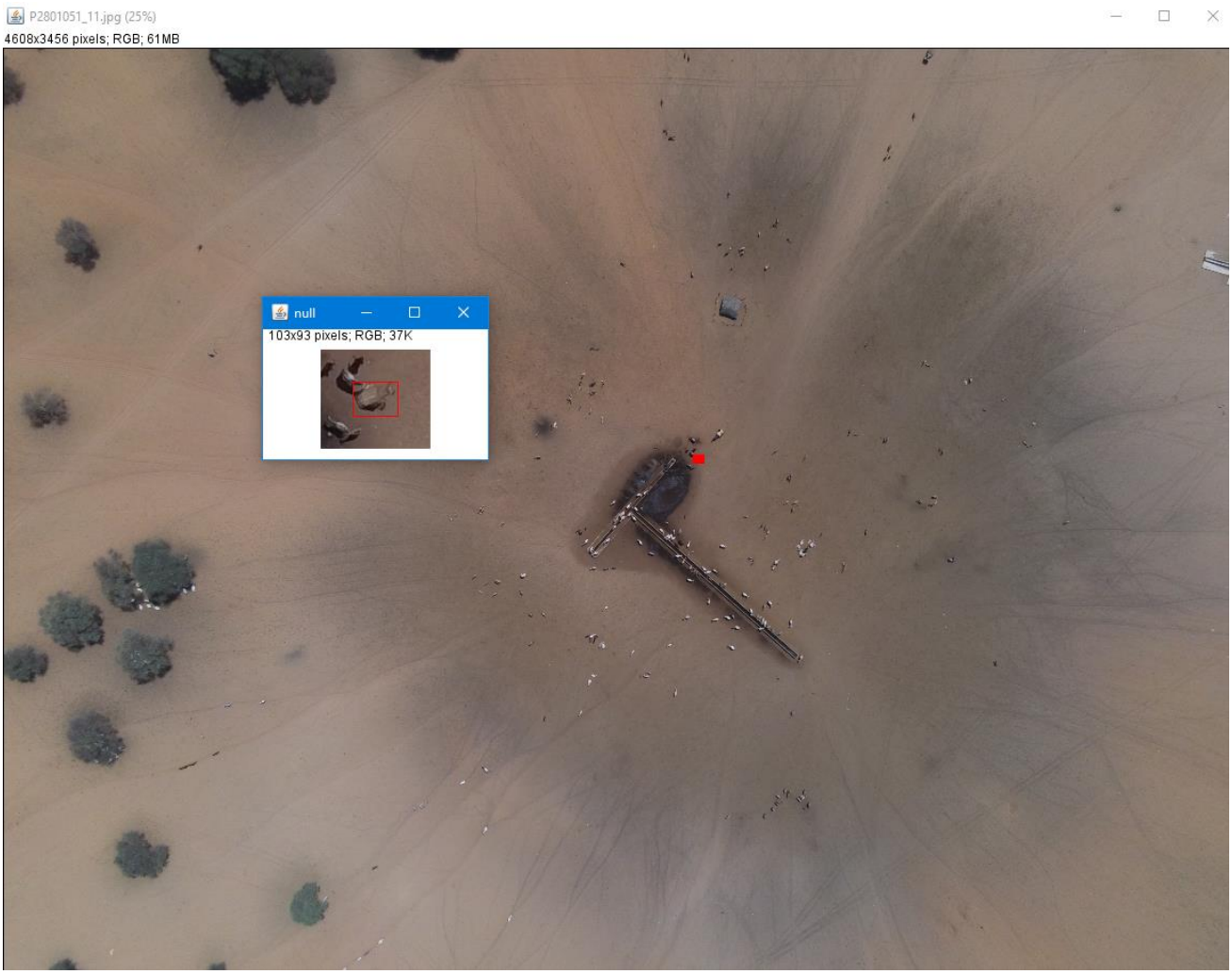
Images block

Allows to navigate through the images in the selected folder in the Input/Output block. The name of the current image is displayed, in blue if already seen, in red if never seen. Changing an image causes the changes made to the current image to be saved.

Displaying

Checker is a visual control tool: it displays the image being analyzed and the selected object in the form of a thumbnail, materialized in the image by a red rectangle.

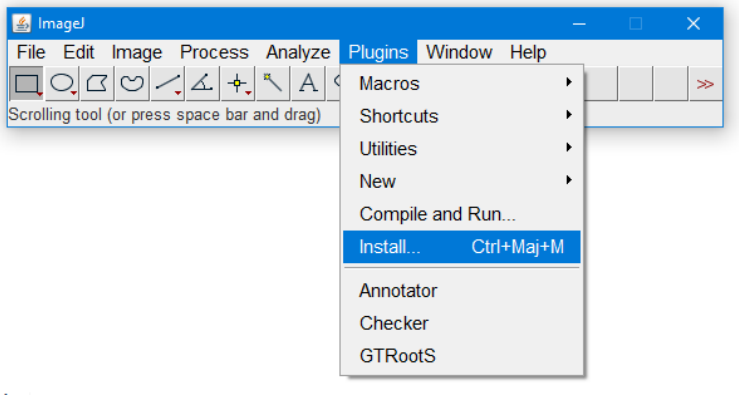
Triggered operations (change of type or class, deletion) are applied on the thumbnail object.



Downloading / Installation

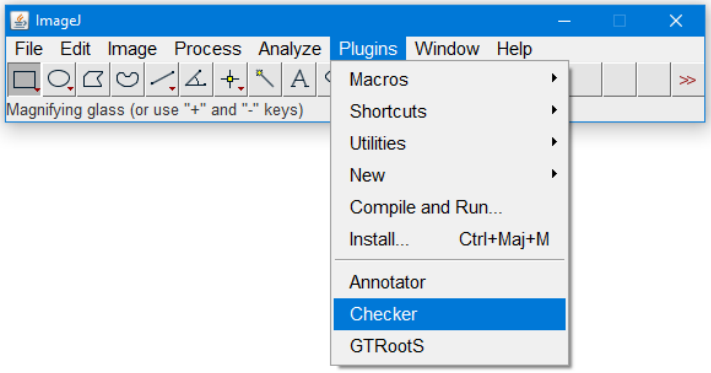
Prerequisite: Install the **ImageJ or Fiji** application on the computer.

Once downloaded to the computer, the **Checker_.jar** plugin must be copied to the plugins subfolder in the root directory of **ImageJ or Fiji** or installed using the GUI of those applications. In both cases, the applications will have to be restarted.



Starting

The Checker plugin is launched from the graphical interface of ImageJ or Fiji.



Citation

Borianne, P., 2020. http://amap-dev.cirad.fr/projects/checker/wiki