



2020 P. Borianne / CIRAD  
Version 1.0 – November 2020

---

Checker is based on the Java application ImageJ developed by Wayne Rasband and distributed under a CeCILL-B license.

---

## 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.

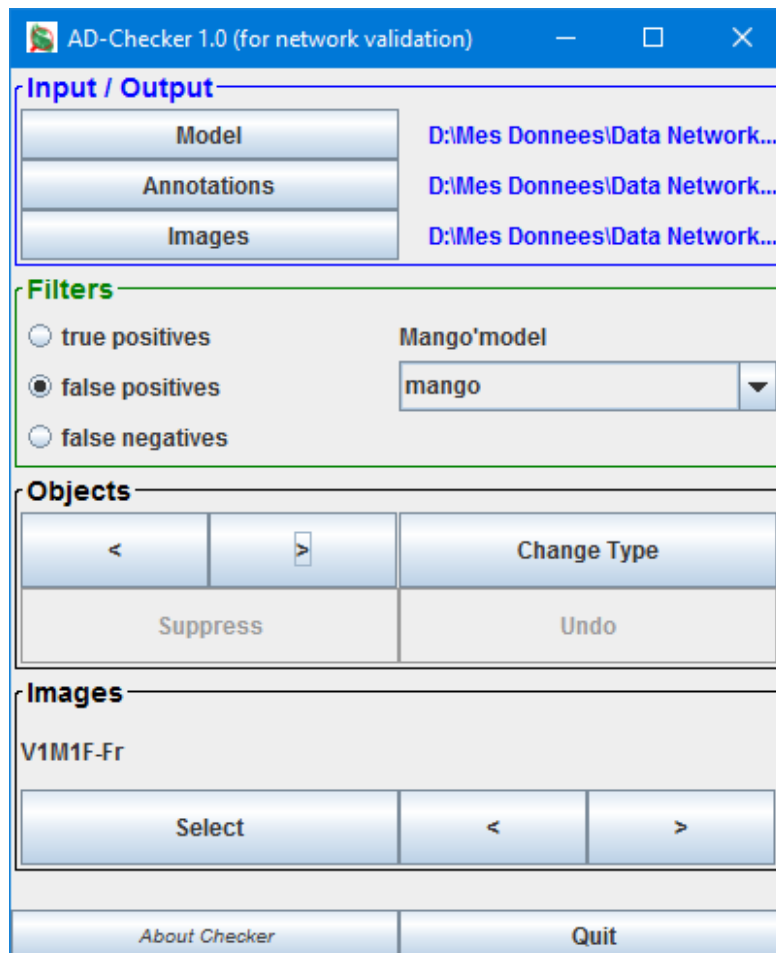




Figure 1 : IHM

## 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 MANGOES
# the color is defined by the rgb string where r, g and b are respectively the red,
green and blue componantes ;
# these components range from 0 to 255
#
MODELNAME  Mango
OBJNAME    mango
OBJCOLOR   255 145 35
OBJNAME    other
OBJCOLOR   35 145 255
```



- The **Annotations** folder: it contains TXT files with exactly the same radical as the images contained in the Images folder.

ID	BX	BY	Width	Height	Name	Type
1	1823	480	43	18	other	FP
2	411	895	23	37	other	FP
3	646	868	32	50	mango	FN
4	1770	723	28	50	other	FP
5	371	1145	52	55	mango	TP
6	146	1164	45	57	mango	TP
7	422	1387	41	63	mango	TP
8	348	1283	43	49	other	FP
9	356	1359	50	60	mango	FN
10	394	1241	40	56	mango	TP
11	325	1190	32	51	mango	TP
12	356	1221	44	56	mango	TP

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

### Filters block

The filters used are specified here in terms of Type ("True Positive", "False Positive" or "False Negative") and class. In Figure 1 it is indicated on the one hand that only the False Positives will be "seen" and on the other hand that in case of modification they will be assigned to the mango 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. Changing an image causes the changes made to the current image to be saved.

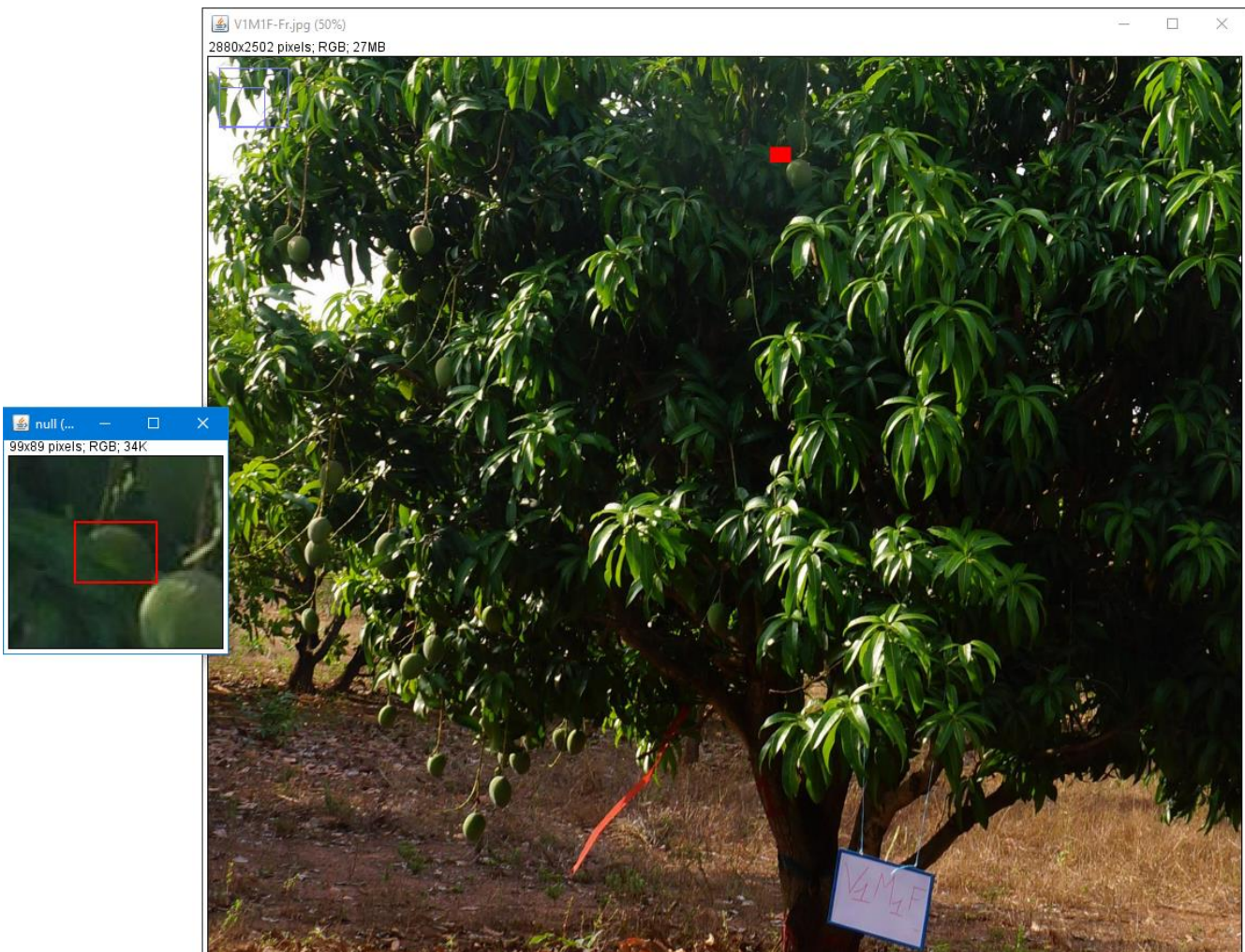
Some values of the TXT files will be modified.



ID	BX	BY	Width	Height	Name	Type
1	1823	480	43	18	other	FP
2	411	895	23	37	other	FP
3	646	868	32	50	remove	RM
4	1770	723	28	50	other	FP
5	371	1145	52	55	mango	TP
6	146	1164	45	57	mango	TP
7	422	1387	41	63	mango	TP
8	348	1283	43	49	mango	TP
9	356	1359	50	60	mango	FN
10	394	1241	40	56	mango	TP
11	325	1190	32	51	mango	TP
12	356	1221	44	56	mango	TP

### 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.



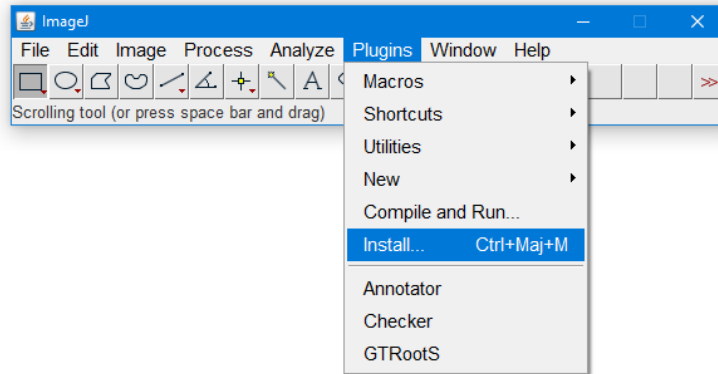
Triggered operations (change of type or class, deletion) are applied on the thumbnail object.



## Downloading / Installation

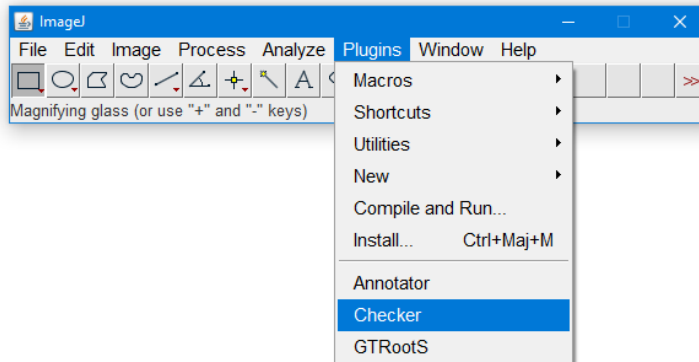
Prerequisite: Install the **ImageJ** or **Fiji** application on the computer.

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>