



2020 P. Borianne / CIRAD
Version 1.0 - Novembre 2020

Checker s'appuie sur l'application Java ImageJ développée par Wayne Rasband et distribuée sous licence CeCILL-B.

Guide Utilisateur

Checker a été développé dans le cadre du projet AgroDeep (AD) ; il permet d'analyser visuellement, et le cas échéant de corriger, les erreurs de prédictions d'un réseau de neurones spécialisé. Il est distribué sous licence Cecill-B.

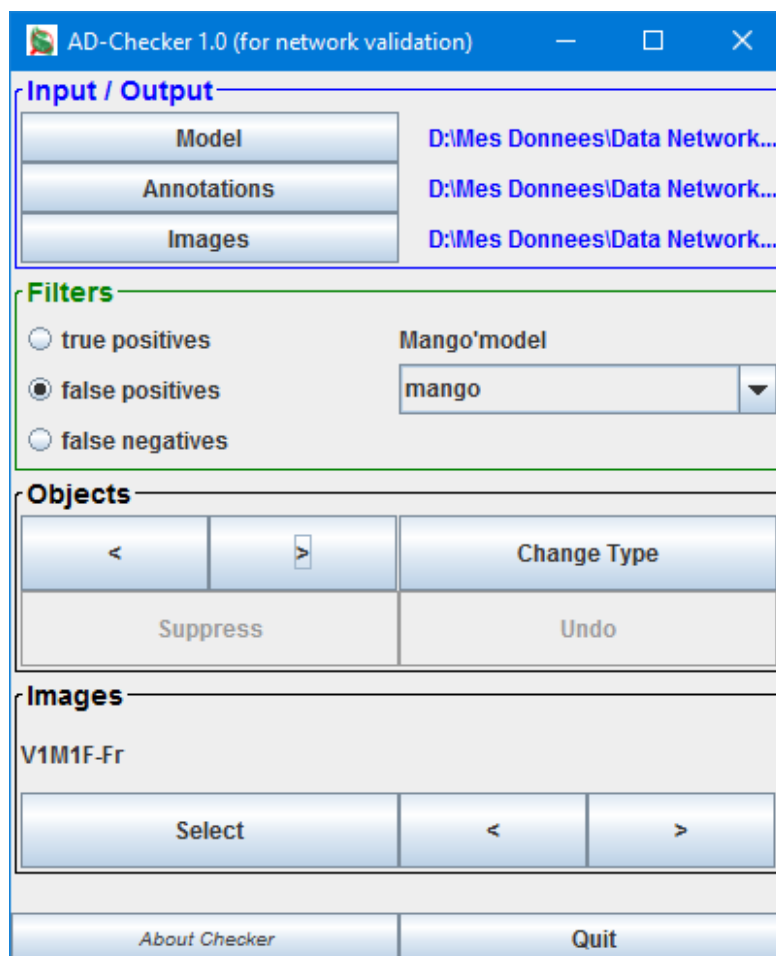




Figure 1 : IHM

Principe

La validation d'un réseau consiste entre autres à comparer les annotations de l'expert aux prédictions du réseau. Ce travail conduit à répartir les objets en trois types distincts :

- Les Vrai Positifs, constitués des objets simultanément annotés par l'expert et détectés par le réseau,
- Les Faux Positifs, correspondant aux objets détectés par le réseau mais ne correspondant à aucune annotation experte,
- Les Faux négatifs, correspondant aux objets annotés par l'expert mais pas détectés par le réseau.

Le contrôle des résultats consiste à s'assurer que les différents objets présents dans l'image sont correctement « typés » ; il est nécessaire de pouvoir corriger les erreurs n'incombant pas au réseau (par exemple des FAUX Faux Positifs qui correspondent à un défaut d'annotation experte).

Le plugin Checker permet de visualiser rapidement les éléments par type et si nécessaire

1. de transférer un objet du type Faux Positif au type Vrai Positif, corrigeant par là-même un défaut d'annotation experte,
2. de supprimer un objet du type Faux Négatifs, corrigeant par là-même une annotation par excès,
3. de reclasser un objet, notamment dans le cas de réseaux multi-classes

Bloc Entrée / Sortie (Input / Output)

Sont ici spécifiés le modèle de données à utiliser et les dossiers contenant respectivement les images et les fichiers de type.

- Le fichier modèle à utiliser : c'est un fichier TXT qui spécifie le nom et la couleur des classes d'objets présents dans les images.

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



- Le dossier *Annotations* : il contient des fichiers TXT ayant rigoureusement le même radical que les images contenues dans le dossier *Images*

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

- Le dossier *Images* : il contient des fichiers Image (png, jpg, JPEG,...) ayant rigoureusement le même radical que les fichiers contenus dans le dossier *Annotations*

Bloc Filtres (Filters)

Sont ici spécifiés les filtres utilisés en terme de **Type** (« Vrai Positifs », « Faux Positifs » ou « Faux Négatifs ») et de **classe**. Dans la Figure 1 est indiqué d'une part que seront « vus » seulement les Faux Positifs et qu'en cas de modification, ils seront affectés à la classe *mango*.

Bloc Objets (Objects)

Sont ici proposées les fonctionnalités applicables aux objets :

- < : accéder à l'objet précédent de l'image courante dans le type spécifié,
- > : accéder à l'objet suivant de l'image courante dans le type spécifié,
- **Change Type** : permet selon le cas de modifier le type ou la classe de l'objet courant,
- **Suppress** : permet de supprimer l'objet courant de la liste des objets associé à l'image courante,
- **Undo** : annule la dernière opération réalisée.

Les fonctionnalités sont bloquées / débloquées selon les filtres utilisés.

Bloc Images

Permet de naviguer au sein des images du dossier sélectionné dans le bloc Entrées / Sorties. Le changement d'image entraîne la sauvegarde des modifications réalisées sur l'image courante.

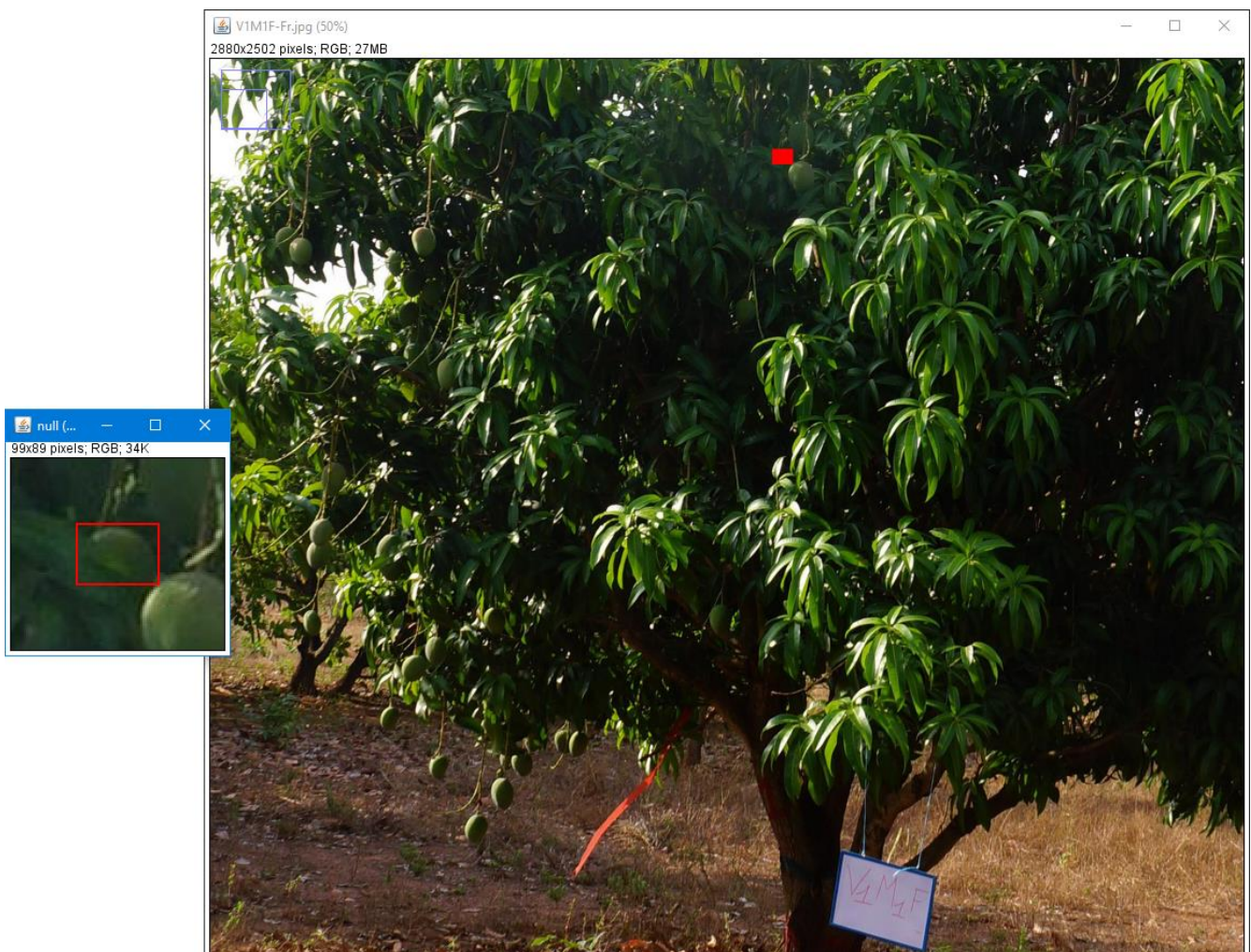
Certaines valeurs des fichiers TXT seront ainsi modifiées.



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

Visualisation

Checker est un outil de contrôle visuel : il affiche l'image en cours d'analyse et l'objet sélectionné sous forme de vignette, matérialisé dans l'image par un rectangle rouge.



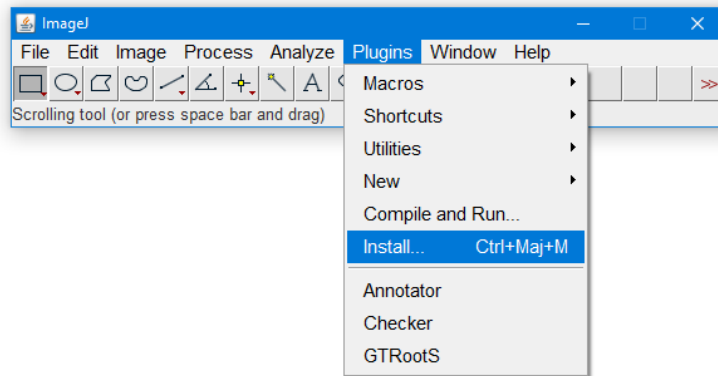
Les opérations déclenchées (changement de type ou de classe, suppression) sont appliquées sur l'objet de la vignette.



Téléchargement / Installation

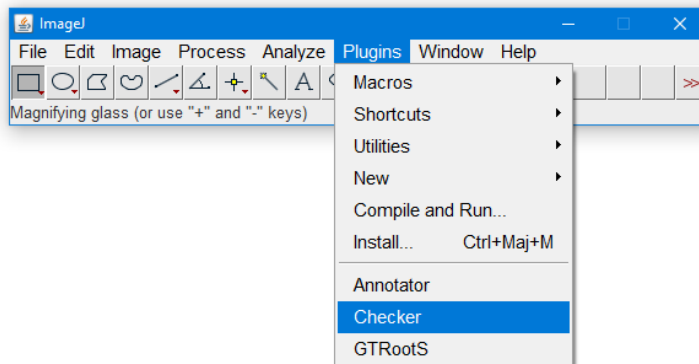
Prérequis : installer l'application ImageJ ou Fiji sur l'ordinateur.

Une fois téléchargé sur l'ordinateur, le plugin **Checker.jar** doit être copié dans le sous-dossier **plugins** du répertoire racine d'ImageJ ou de Fiji ou installer à l'aide de l'interface graphique des dites applications. Dans les deux cas, ces dernières devront être relancées.



Lancement

Le plugin Checker est lancé à partir de l'interface graphique d'ImageJ ou de Fiji.



Citation

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