

# Détecteur multi-objets dans des images naturelles exploitant un graphe de relations spatiales

## Document de tests

# Sommaire

Classe RelationshipGraphGenerator .....	3
Fiche de test de la fonction « getWhitelistCsv » .....	3
Fiche de test « FonctionGlobalClasse » .....	4
Classe YoloAnnotationFilesGenerator .....	5
Fiche de test de la fonction « generateAnnotsYolofile » .....	5

# Classe RelationshipGraphGenerator

Fiche de test de la fonction « getWhitelistCsv »

RelationshipGraphGenerator		getWhitelistCsv
----------------------------	--	-----------------

## DESCRIPTION

Projet :	N° de fiche :	Auteur :	Référence séquence de test :
RelationshipGraphGenerator.py	1	CHENY Julien	getWhitelistCsv_NormalWork
Date du test :			
20/03/2019			

## IDENTIFICATION DU CAS

La classe « RelationshipGraphGenerator.py » contient une méthode pour récupérer une liste d'un fichier csv.

## DESCRIPTION DU TEST (valeurs, actions)

Exécution de la fonction avec en paramètre :

- paramètre 1 : « whitelist.csv »

Contenu du fichier « whitelist.csv » : « obj1,obj2,obj3,obj4 »

## RESULTATS ATTENDUS

Une liste de strings contenant :  
[obj1, obj2, obj3, obj4]

## RESULTATS OBTENUS

Test réussi

Anomalie n° :

## Fiche de test « FonctionGlobalClasse »

RelationshipGraphGenerator	FonctionGlobalClasse
----------------------------	----------------------

DESCRIPTION			
Projet :	N° de fiche :	Auteur :	Référence séquence de test :
RelationshipGraphGenerator.py	1	CHENY Julien	FonctionGlobalClasse
Date du test :			
20/03/2019			

IDENTIFICATION DU CAS
La classe « RelationshipGraphGenerator.py » contient une méthode d'initialisation qui génère un graphe (bibliothèque NetworkX) de relations spatiales entre objets depuis des fichiers d'annotations d'images (bibliothèque Pycocotools, base d'images Coco)
L'objectif est de tester cette méthode.

DESCRIPTION DU TEST (valeurs, actions)
Vérifier que la fonction « init » génère bien un graphe dans la variable « g ». Avec pour paramètres : <ul style="list-style-type: none"><li>- paramètre1 : « whitelist_test.csv »</li><li>- paramètre2 : « /srv/datas/annotations/instances_train2014.json »</li></ul> Le fichier « whitelist.csv » contient : « person,car,bicycle » Le fichier « instances_train2014.json » provient du site cocodataset.org

RESULTATS ATTENDUS
Arcs sortant du nœud 1 (person) : (1, 1, {'object': {'up': 101441, 'down': 100761, 'left': 575236, 'right': 575236}}) (1, 3, {'object': {'up': 11987, 'down': 5172, 'left': 52858, 'right': 53011}}) (1, 2, {'object': {'up': 762, 'down': 6392, 'left': 11803, 'right': 12040}}) Le graphe dispose aussi de six arcs supplémentaires.

RESULTATS OBTENUS
Test réussi
Anomalie n° :

# Classe YoloAnnotationFilesGenerator

Fiche de test de la fonction «generateAnnotsYolofile»

YoloAnnotationFilesGenerator		generateAnnotsYolofile
------------------------------	--	------------------------

## DESCRIPTION

Projet :	N° de fiche :	Auteur :	Référence séquence de test :
YoloAnnotationFilesGenerator.py	1	CHENY Julien	generateAnnotsYolofile _NormalWork
Date du test :			
20/03/2019			

## IDENTIFICATION DU CAS

La classe « YoloAnnotationFilesGenerator.py » contient une méthode generateAnnotsYolofile. Cette fonction va générer les fichiers d'annotations des objets de chaque image pour l'apprentissage du réseau de neurones Yolo.

## DESCRIPTION DU TEST (valeurs, actions)

Exécution de la fonction « test » avec en paramètre :

- paramètre 1 (configfile) : « test/configs\_test.conf »

Contenu du fichier « configs\_test.conf » : fichier de configuration de tests

## RESULTATS ATTENDUS

Génère des fichiers dans le répertoire « /srv/datas/ »  
Les dossier person, car et bicycle.

## RESULTATS OBTENUS

Test réussi

Sortie standard :

forget node : car (2) with 6.17%

Resultat :

Id des objets selectionnes :

[1, 3]

Noms des objets selectionnes :

['person', 'bicycle']

Anomalie n° :