

# PACT - groupe 3.2

## Module de détection de déchets

### Rapport PAN 2

Victor Masiak  
Nicolas Jow

19 janvier 2020

## 1 Objectif du module

L'objectif de notre groupe PACT est de mettre au point un robot éducatif qui, au travers d'activités pédagogiques, sensibilisera les enfants au tri sélectif. Pour ce faire, il devra être capable de repérer et classer les déchets présents dans son environnement, à l'aide d'une caméra numérique.

L'objectif de ce module est alors la mise au point d'un système d'intelligence artificielle permettant de transformer les données brutes de la caméra (c'est-à-dire des images) en données exploitables par le robot (c'est-à-dire les positions relatives des déchets par rapport au robot, ainsi que leur type).

## 2 Ce que l'on a fait

Nous avons tout d'abord commencé par nous documenter sur les réseaux de neurones en général, sur internet. Nous avons compris le fonctionnement d'un neurone, comment plusieurs neurones peuvent être reliés pour former un réseau neuronal multicouche, les différents algorithmes d'entraînement, etc. Nous avons rapidement compris que nous allions avoir besoin de récupérer un réseau pré-entraîné, pour ensuite le ré-entraîner sur des images appropriées. Nous nous sommes donc mis à la recherche d'un tel réseau.

Nous sommes tout d'abord tombé sur un projet assez complet<sup>1</sup>, contenant une implémentation du réseau Mask-RCNN ainsi qu'une base de données de divers débris dans la nature. Nous avons essayé de comprendre son fonctionnement et de le tester, mais nous avons finalement abandonné cette piste pour deux raisons :

- Nous avons trouvé très peu de documentation sur internet. Hors, nous avons rencontré de multiples problèmes d'installation des différentes dépendances, que nous n'avons pas réussi à résoudre complètement.
- Le projet est très complet, et donc déjà fini et fonctionnel. Par conséquent, nous n'aurions eu que très peu de travail, celui-ci se limitant à télécharger le projet et ses dépendances. L'apport pédagogique aurait donc été très limité.

Nous avons alors trouvé le réseau Inception V3, pré-entraîné sur la banque d'images ImageNet. Après nous être documenté sur internet, il nous a semblé que ce réseau était très performant, et surtout adapté à notre situation. Nous avons donc cherché une base d'images de déchets pour pouvoir ré-entraîner la dernière couche du réseau. Nous avons trouvé une base d'images sur un

---

1. <https://pythonawesome.com/a-growing-image-dataset-of-waste-in-the-wild/>

challenge kaggle<sup>2</sup> disposant de six classes : carton, verre, métal, papier, plastique et non recyclable. Cette base de données nous a semblé correspondre parfaitement à nos besoins, mais nous avons cependant des doutes sur sa taille (environ 500 images par catégorie, ce qui nous semblait peu). Nous avons donc suivi un tutoriel du site de tensorflow<sup>3</sup> pour ré-entraîner la dernière couche du réseau avec notre base de données. L'évolution de l'entraînement est présentée dans la figure 1 ci-dessous :

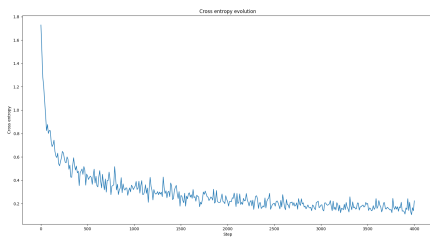


FIGURE 1 – Ré-entraînement de la dernière couche du réseau Inception V3 sur notre BDD

Après entraînement, le taux d'erreurs du réseau est d'environ 10%. Ce premier résultat est satisfaisant, surtout si on prend en compte la complexité de certains déchets, qui peuvent être interprétés de différentes manières (par exemple des feuilles de magazines plastifiées, que l'on pourrait classer comme du papier mais aussi comme du plastique). Nous avons de plus repéré quelques images mal labellisées. Nous espérons néanmoins réduire ce taux d'erreurs dans le futur.

Nous nous sommes ensuite intéressés aux cas où l'image contient plusieurs déchets, qui ne sont pas centrés sur celle-ci. Nous avons pour cela choisi de balayer l'image, et d'appliquer le réseau sur des portions extraites de l'image, pour sélectionner celles

qui produisent une réponse forte de l'une des classes. Puisque, le réseau passe plusieurs fois sur chaque déchets (à différentes échelles, de manière plus ou moins centrée, etc.), il faut donc ensuite trier les coordonnées retournées. Nous avons pour cela décidé d'effectuer un barycentre des coordonnées retournées pour chacune des classes. Cette méthode nous a donné des résultats plutôt satisfaisants, mais présente deux inconvénients majeurs :

- Le temps d'exécution : en effet, si un appel du réseau se fait relativement rapidement, nous l'appelons ici plusieurs centaines de fois.
- Le manque d'adaptabilité : puisque nous effectuons un barycentre pour chacune des classes, on ne peut détecter qu'un seul déchet de chaque type sur l'image fournie.

Voici ci-dessous (figure 2) les résultats pour une image avec :

- 1 déchet type plastique
- 1 déchet type verre
- 2 déchets type carton

Les déchets détectés sont représentés par un point noir. Le type détecté n'est pas présenté ici, mais est correct pour tous les déchets de cette image.



FIGURE 2 – Test du réseau

A titre indicatif, nous avons fait tourner l'algorithme sur un CPU i7, et cela a pris environ 4 minutes et 15 secondes.

2. <https://www.kaggle.com/asdasdasasdas/garbage-classification/data>

3. [https://www.tensorflow.org/hub/tutorials/image\\_retraining](https://www.tensorflow.org/hub/tutorials/image_retraining)

### 3 Ce que nous allons faire maintenant

Dans les semaines qui vont suivre, nous allons nous concentrer sur plusieurs points :

- Tout d’abord, nous allons chercher à exploiter la première sortie du réseau Inception V3, afin d’améliorer et d’accélérer la détection sur une image avec plusieurs déchets non centrés.
- Nous allons ensuite chercher à rajouter une étape au ré-entraînement du réseau, où nous défreezerons toutes les couches.
- Nous chercherons aussi à agrandir notre BDD, soit en trouvant une nouvelle avec plus d’images, soit en développant un scraper pour google images.
- Enfin, nous continuerons d’explorer d’autres options pour le choix de l’architecture du réseau, Inception n’étant pas vraiment connu pour être rapide. Le réseau Mobilenet est par exemple apparu a de multiples reprises dans nos lectures.