

SYS843-01 RÉSEAUX DE NEURONES ET SYSTÈMES FLOUS (H2019)  
DÉPARTEMENT DU GÉNIE DE LA PRODUCTION AUTOMATISÉE

---

## DÉTECTION D'OBJETS

---

Soumis par  
Danny SAUVAL  
Département du Génie de la Production Automatisée  
École de Technologies Supérieures  
Montréal, QC

Soumis à  
Ismail BEN AYED  
École de Technologies Supérieures  
Montréal, QC



Le génie pour l'industrie

# 1 Mise en situation

Le fonctionnement de l'être humain est fascinant et passionnant. L'intelligence dont nous faisons preuve dès nos premiers jours nous permet de rapidement prendre conscience du monde qui nous entoure et de le comprendre. La détection d'objet est un bon exemple. Après avoir vu quelques fois un objet, nous sommes capables de facilement le détecter et de le reconnaître quand on le voit, et cela, même si l'objet est différent.



**FIGURE 1:** À gauche une chaise classique [1] et à droite une chaise de designer [2]

Mes propos peuvent être illustrés par la Figure 1. À gauche, on a une chaise classique : un dossier et un plateau supporté par 4 pieds. À droite, on a une chaise de designer qui ne correspond pas du tout à la définition d'une chaise. Pourtant notre cerveau est capable de l'identifier comme tel.

Le but de mon projet est d'entraîner un réseau de neurone à être capable de détecter et reconnaître des objets comme notre cerveau le fait.

Les applications de ce type de systèmes sont très diverses, on peut par exemple citer l'utilité pour une voiture autonome qui pourrait reconnaître les objets présents sur la route et son environnement. Le système pourrait aussi être utilisé dans un robot d'assistance à la personne qui pourrait aller chercher un objet dans la maison.

Le but du projet est dans un premier temps d'être capable de détecter un objet dans une image (et plus tard une vidéo), et si les résultats sont concluants ajouter au système la possibilité de détecter plus d'objets.

## 2 Problématique abordée

La détection d'objet est un sujet largement abordé dans la littérature. La grande majorité des algorithmes sont basés sur l'utilisation d'un CNN, qui permet généralement de faire de la classification d'image. Si on prend en photo un chat, l'algorithme sera capable de classer l'image comme étant celle d'un chat. Cependant, dans ce projet on ne veut pas faire de classification, on veut identifier dans une image plusieurs objets. L'idée de base est donc de découper l'image de base en plusieurs régions, et d'envoyer chacune de ces régions à un CNN.

Après un rapide survol de la littérature, voici les algorithmes les plus utilisés :

1. **Faster R-CNN** (Region-Convolution Neural Network) : Faster R-CNN utilise un second réseau de neurones qui prédit les régions d'intérêt. Celles-ci sont ensuite envoyées à un CNN qui les

classifie.

2. **You Only Look Once (YOLO)** : Cette technique utilise un seul CNN qui prédit la région d'intérêt et la classe associée à celle-ci.
3. **SSD : Single Shot MultiBox Detector** : Cette technique est basée sur l'utilisation d'un seul CNN dont le réseau est parcouru du début à la fin une seule fois (Single Shot). Il est combiné à une technique de régression de région d'intérêt (MultiBox).

Un survol de littérature pourra permettre de détailler ces méthodes et de voir quelle approche permet de mieux répondre à notre problème.

### 3 Objectifs du projet

L'objectif du projet va être d'entraîner un réseau de neurones à détecter et reconnaître un objet dans une image. Par exemple, 80% d'une database contenant des vélos va être utilisé pour entraîner un réseau. On testera la précision de celui-ci à l'aide des 20% restant de la database. Si le résultat n'est pas suffisant, on entraînera à nouveau le réseau. On pourra éventuellement utiliser un réseau pré-entraîné tel que AlexNet ou MobileNet. Notre algorithme devra être capable d'apprendre à détecter et reconnaître de nouveaux objets. Il faudra également optimiser l'algorithme pour essayer de détecter et reconnaître les objets en temps réel. Cela permettrait de rendre l'algorithme compatible avec une voiture autonome, qui a besoin d'informations en temps réel.

### 4 Méthodologie

La première étape du projet va être de se documenter sur les réseaux de neurones convolutifs afin de bien comprendre leur fonctionnement. En effet, ils sont la base des techniques brièvement décrites dans la [section 2](#).

Une synthèse de ce qui se fait dans la littérature sera également faite dans l'objectif d'être plus clair sur les méthodes qui sont les plus efficaces. Cela nous permettra de savoir laquelle ou lesquelles sont les plus intéressantes à implémenter.

Les outils que je compte utiliser seront Matlab ainsi que OpenCV combiné à Python ou bien au C++. Je connais bien ces outils ce qui va donc m'éviter d'apprendre à utiliser un nouvel outil.

J'ai trouvé sur internet un site qui centralise plusieurs databases destinées à la détection et la reconnaissance d'objets : <https://www.cs.ubc.ca/~murphyk/Vision/objectRecognitionDatabases.html>. Elles sont gratuitement accessibles et directement téléchargeables.

## Références

- [1] Callum - chaise de salle à manger. <https://www.habitat.fr/p/callum-chaise-de-salle-a-manger-naturel>. Accessed : 01-29-2019.
- [2] Chaise design original pin up par infiniti. <http://www.archiexpo.fr/prod/vitra/product-80422-684808.html>. Accessed : 01-29-2019.
- [3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn : Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd : Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [5] Rohith Gandhi. R-cnn, fast r-cnn, faster r-cnn, yolo — object detection algorithms. <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>. Accessed : 01-29-2019.
- [6] Adrian Rosebrock. Object detection with deep learning and opencv. <https://www.pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/>. Accessed : 01-29-2019.