VA52 - Reconnaissance des formes
YOLO, pix2pix, MonoDepth & Co.

La communauté gravitant autour du deep learning est extrêmement active. Les sources et datasets d'énormément de projet couvrant tout un éventail d'application sont disponibles en ligne. Avant de se lancer dans l'implémentation d'une méthode "from scratch", il est plus que conseillé de prendre connaissances des méthodes de l'état de l'art proposées par la communauté afin de ne pas ré-inventer la roue. Au cours de cette séance, nous nous intéresserons à l'exploitation de trois d'entre-elles : YOLO, pix2pix et MonoDepth.

I. YOLO

YOLO (You Only Look Once) est une méthode de détection d'objets en temps réel au sommet l'état de l'art.

Les méthodes de détection d'objet habituelles utilisent des classifieurs et/ou des localisateurs pour effectuer la détection. Ils appliquent un même modèle à plusieurs endroits de l'image donnée et à différentes échelles. Enfin, les régions de l'image qui ont un score élevé sont considérées comme des détections.

YOLO, quant à lui, alimente directement son réseau de neurones par l'image complète. Le réseau divise lui-même l'image en régions et prédit les limites et les probabilités pour chacune d'entre-elles. Les différentes boîtes englobantes obtenues sont pondérées par les probabilités de prédiction. Cette approche présente plusieurs avantages par rapport aux systèmes basés sur les classifieurs/localisateurs. L'image complète étant examinée au moment de l'inférence, le contexte global de l'image a une répercussion sur les prédictions locales. De plus, les prédictions sont obtenues après une seule inférence du réseau, contrairement à d'autres approches qui peuvent nécessiter des milliers d'évaluations pour une seule image. Par conséquent, YOLO est très rapide et permet d'être exploité pour des applications temps-réel.

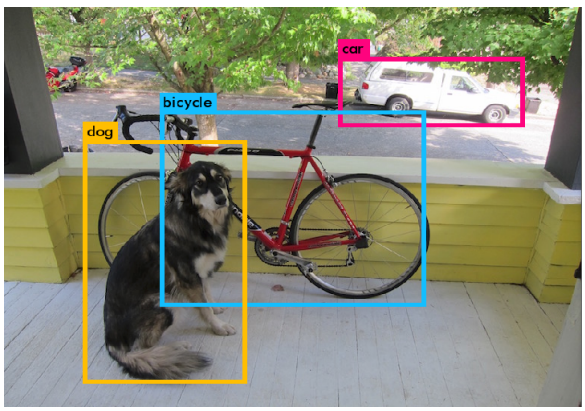

Figure 1: Exemple de détections obtenues avec YOLO pour une image donnée

Exercice : Les sources, les données, les poids et la procédure d'utilisation de YOLO sont disponible à cette adresse :
https://github.com/eriklindernoren/PyTorch-YOLOv3.
Essayer le !

II. pix2pix

Pix2pix est une méthode générale destinée à la conversion ("translation") d'une image source donnée en une image cible. Il s'agit d'un problème difficile qui nécessite généralement l'élaboration d'un modèle spécialisé et d'une fonction de coût spécifique à chaque tâche de conversion recherchée. Pix2pix est basé sur le principe des réseaux adverses génératifs (generative adversarial networks ou GANs). Un GAN est un modèle génératif où deux réseaux, un générateur et un discriminateur sont placés en compétition dans le processus d'apprentissage. Le premier réseau, le générateur, génère une image pour atteindre une image cible, pendant que son adversaire, le discriminateur essaie de détecter si cette image est réelle ou bien si elle est a été créée par le générateur. Une très bonne explication du GAN et de pix2pix est proposée ici : https://affinelayer.com/pix2pix/.

Pix2Pix a été exploité sur une série d'applications différentes telles que :

- Image N&B ⇔ Image RGB
- Photo aérienne ⇔ Carte
- Croquis ⇔ Photo
- Photo de jour ⇔ Photo de nuit
- ...

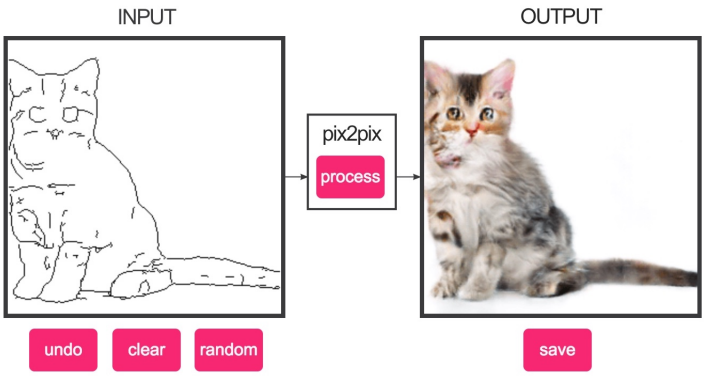

Figure 2: Exemple d'utilisation de pix2pix : Croquis ⇔ Photo

Exercice : Les sources, les données, les poids et les procédures d'utilisation des différentes applications de pix2pix sont disponible à cette adresse :
https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix.
Essayer le pour l'application de votre choix !

III. MonoDepth

Une problématique très étudiée en vision par ordinateur est l'estimation des profondeurs d'une scène. De nombreuses approches ont été proposées, ces-dernières reposent généralement sur le mouvement (Structure-from-Motion), la stéréovision ou encore la géométrie multi-vues. Cependant, la plupart de ces techniques requièrent de multiples observations de la scène. Pour surmonter cette limitation, de nombreux travaux comme MonoDepth

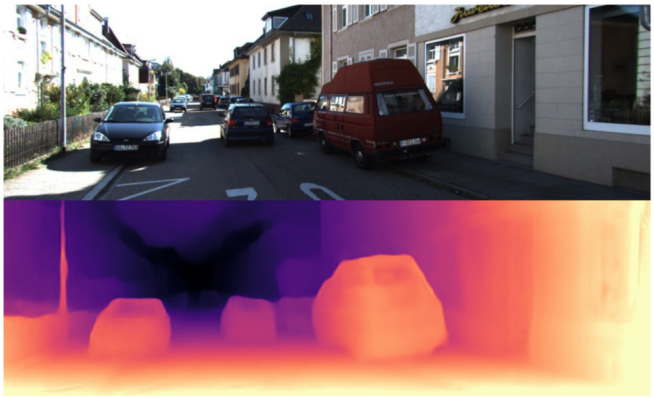

Figure 3: Exemple d'estimation de profondeurs à partir d'une seule image donnée par MonoDepth

cherchent à modéliser l'estimation de la profondeur monoculaire sous le formalisme d'un problème d'apprentissage supervisé.

Exercice : Les sources, les données, les poids et la procédure de MonoDepth sont disponible à cette adresse :
https://github.com/nianticlabs/monodepth2.
Essayer le !