

Le Tri Sélectif

By GELFGED Gary, GRUBER Raphaël, NEJJAR Marouane, POUILLET Teva, RIO Benjamin

<u>Lien GitLab</u>: https://gitlab-cw7.centralesupelec.fr/marouane.nejjar/coding_week_projet_gp9

Sommaire



Description du projet et utilité



Notre organisation, nos sprints, nos écueils



Structure de notre code



Documentations utilisées pour la conception de notre code



Description du projet et utilité

QUOI?

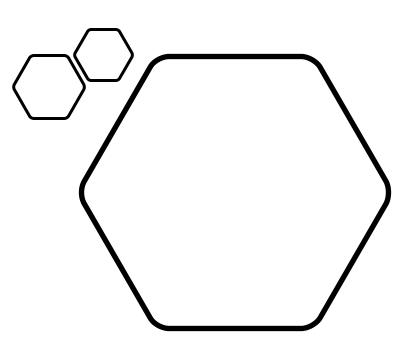
- Un détecteur de déchets qui vous indique sa poubelle
- Utilisable en live avec votre webcam ou en différé avec une image
- Extension : implémentation de notre code dans un robot de tri

POURQUOI?

• Le tri des déchets est l'une des actions de développement durable les plus répandues mais souvent difficile à mettre en œuvre.

POUR QUI?

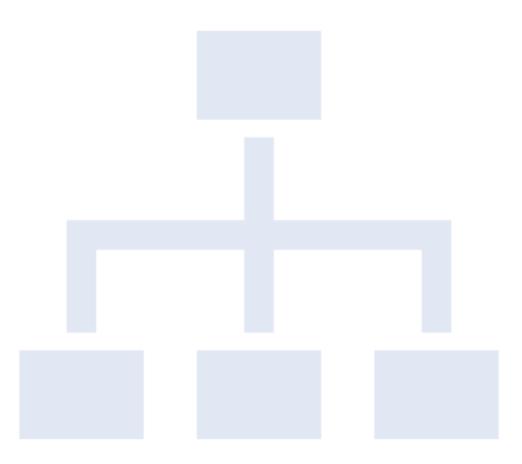
- Un utilisateur dans la vie de tous les jours
- À grande échelle dans des usines de tri





Organisation

Nos sprints, nos écueils



How we did it

SPRINT 2

Utilisation de la méthode HaarCascade

- Collecte des données grâce à l'API
 - Création de nos fichiers XML
- Détection des déchets sur une image
- Détection des déchets sur l'image de la webcam
 - Amélioration HaarCascade

SPRINT 0

Mise en place de la base du projet

- Socle Technique: GIT
- Analyse des besoins
- Moment de réflexion

Adaptation des modules de la semaine 1

- Lancement de la caméra + prise d'une photo
 - Traitement des images

SPRINT 1

Reconnaissance des matériaux

- Création du réseau convolutif sur une base de ResNet
- Entrainement du réseau convolutif sur Garbage Classification
 - Affichage d'un message qui associe les matériaux aux poubelles

SPRINT 3

How we did it

SPRINT 4

Finalisation du MVP

- Agencement de notre code
- Création du *main* notre programme
 - Création de deux propositions :
 - Utilisation de la caméra ;
 - Utilisation d'une image existante.

MAINTENANT TRIONS LES DÉCHETS

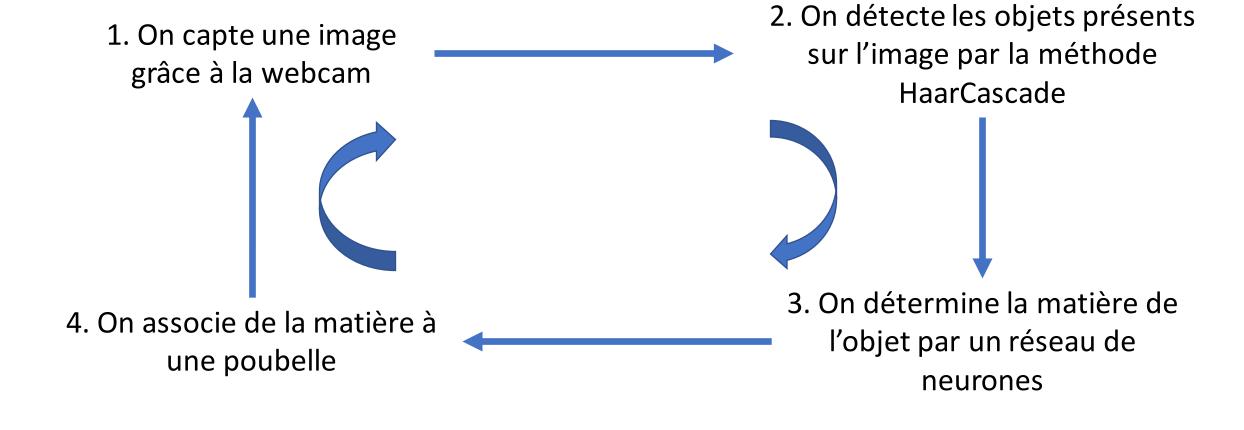


Amélioration du MVP

- Création d'une interface WebApp
- Analyse des performances de nos modèles de détection
 - Amélioration de la détection

SPRINT 5

Résumé du fonctionnement de notre code





Focus: Sprint 2



Sprint 2 : Pas le chemin le plus simple



Facilité

Il existe des modèles tout faits grâce au module *ImageAi* (*YOLO*) mais nous avons décidé de suivre **notre propre démarche**.



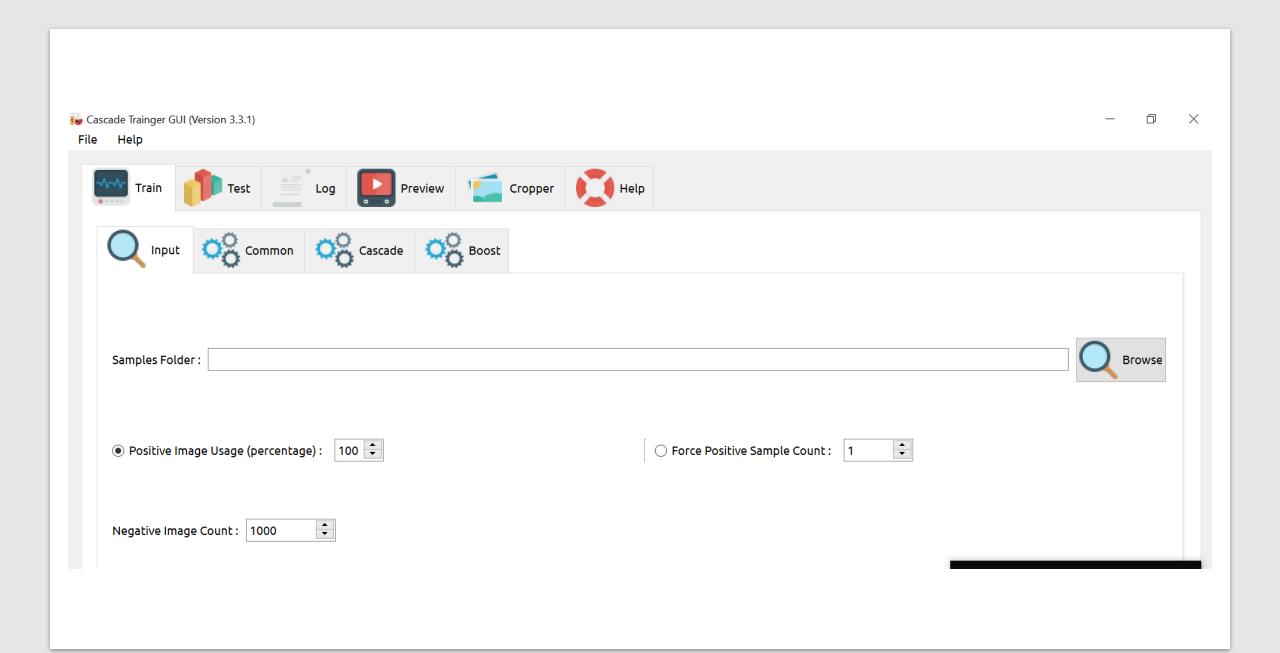
Notre idée

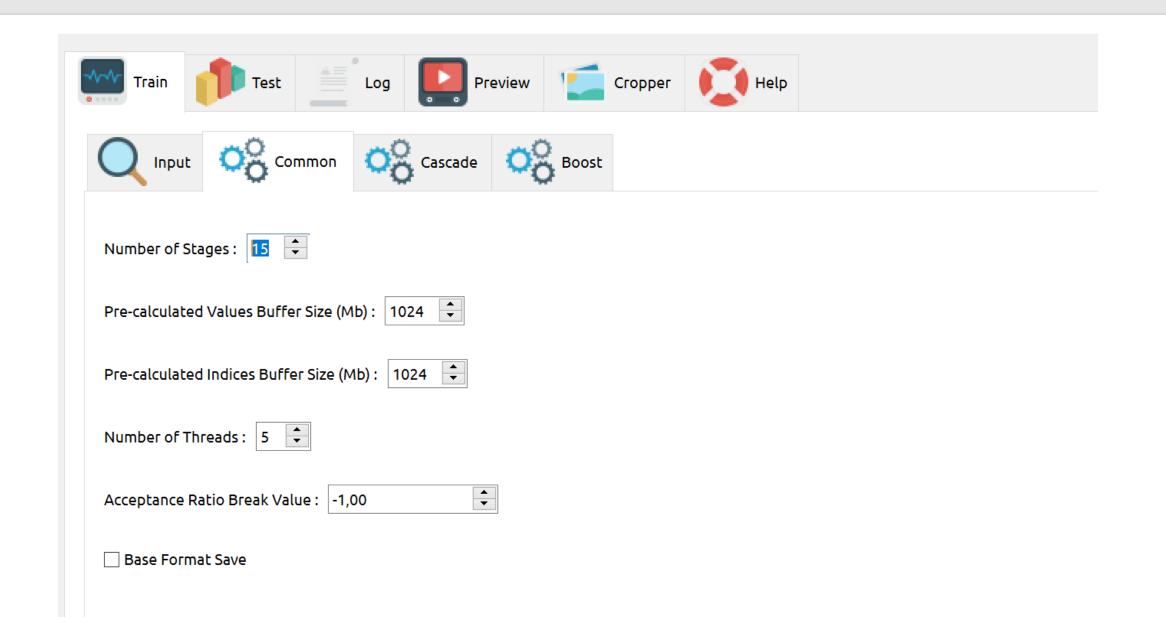
Adapter la méthode HaarCascade pour détecter les objets sur une image et extraire un zoom de l'objet.



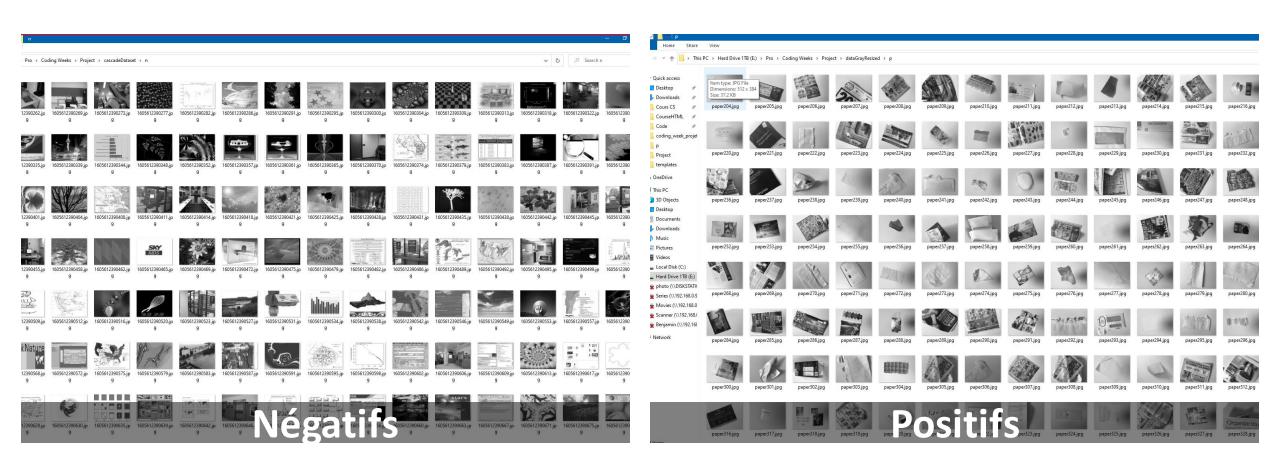
Résolution

Création de **nos propres fichiers XML entrainés sur des milliers d'images** négatives et positives grâce à *CascadeTrainerGui*. (cf. Images sur les dia suivantes)





Aperçu des samples



Sprint 2 : Amélioration de la détection HaarCascade



Suite à la construction de notre fichier XML, le groupe s'est rendu compte d'un défaut lors de la détection. Celle-ci n'étant pas optimale, il arrive que le programme pense qu'un objet est constitué de deux objets. **MAIS l'objet est tout de même localisé!**



Notre idée

Implémenter une fonction fusionRectangle() qui renvoie l'enveloppe convexe rectangulaire des rectangles dessinés sur l'objet (cf. Images à la dia suivante)







Focus: Sprint 3



Sprint 3 : Prendre un modèle et l'entrainer

Comment avons-nous construit notre modèle?



Nous avons utilisé la base de données *Garbage Classification* et le module *Pytorch* pour réaliser le Model.

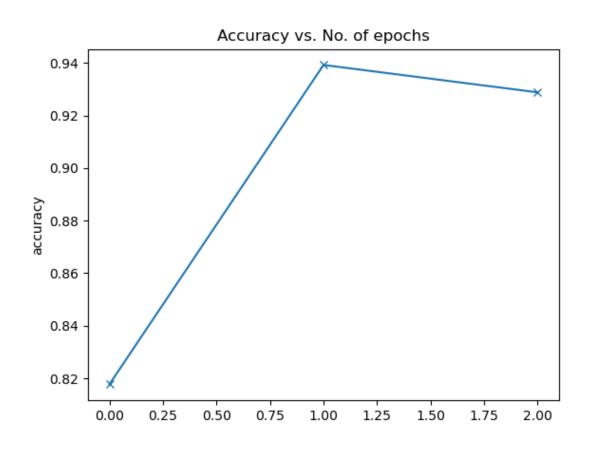


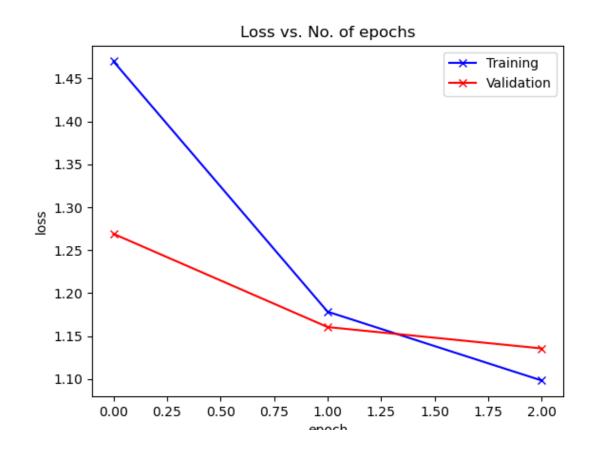
Le Model a été construit grâce à la base *ResNet* et entrainé sur la base de données lors de 3 epochs.



Nous l'avons sauvegardé afin de le réutiliser sans avoir à attendre 1h30

Sprint 3 : Nos résultats





Sprint 3 : De nombreuses difficultés



Problèmes

- Nous avions essayé d'utiliser un model déjà entrainé à la classification de déchet, utilisant tensorflow et openCV mais les résultats étaient très mauvais.
- 2. Impossibilité de lancer le code sur le GPU malgré 4h d'installations de divers drivers et de recherche Google.



Solutions

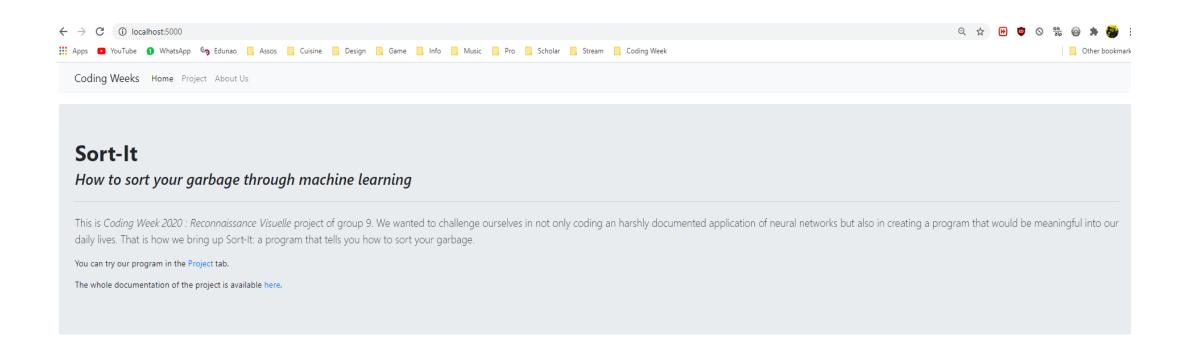
- Nous avons opté pour un autre model que nous avons dû entrainer nousmême.
- 2. Faire tourner les programmes sur le CPU (mais au prix d'un temps de calcul conséquent).



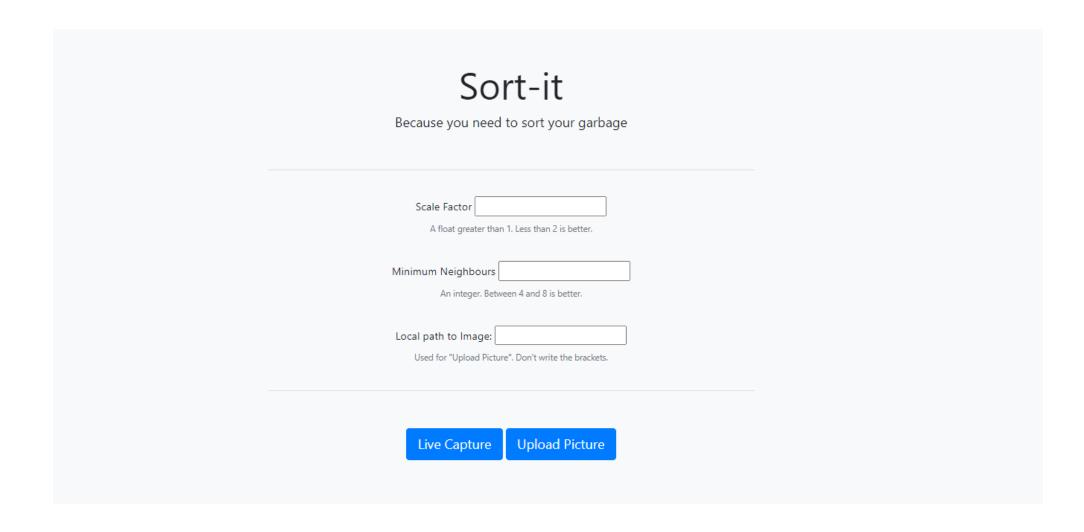
Focus: Sprint 5



Interface graphique : WebApp



Démonstration à la fin

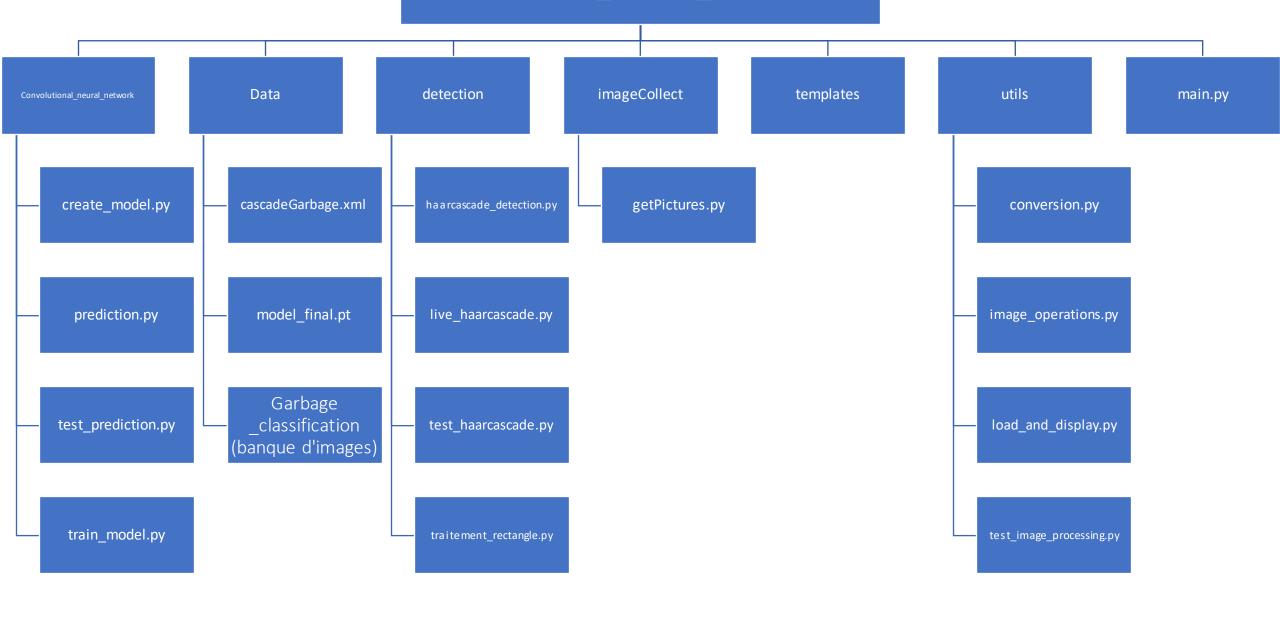


Qualité du code

Structure, modularité



Coding_weeks_2020



Les tests présents dans notre code







Sur la qualité de détection du haarcascade



Sur la distinction du matériau

Documentations





• Guide to make custom HaarCascade File (example with cats on medium.com)

• Learning from our mistakes (Gargabe classification model VGG16) and succeed with harder training (Garbage segregation on Pytorch)

• And obviously: OpenCv library, Numpy library, PyTorch

