


POC OUTIL LABELLISATION IA

Contexte IGN sur labellisation de données IA

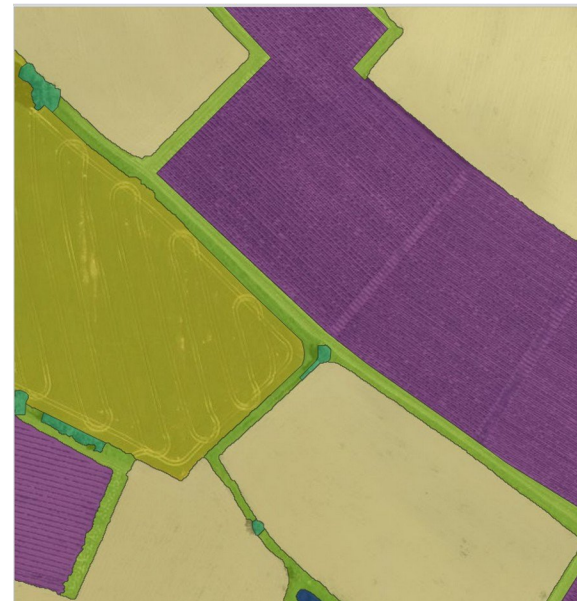
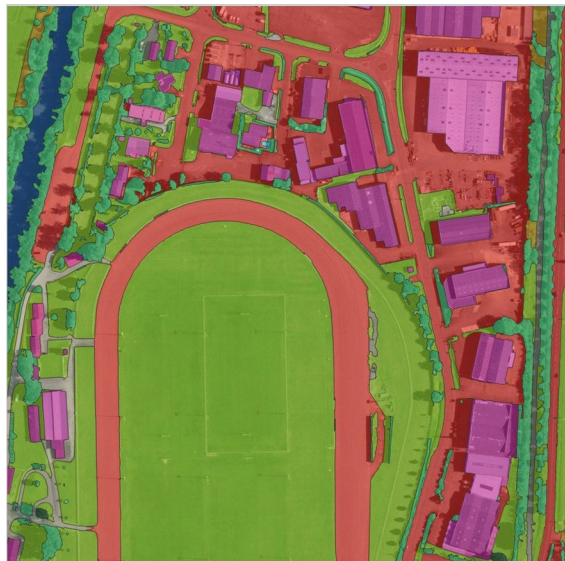
- Besoin IGN : Pas seulement images aériennes mais cartes anciennes.
- Contexte IGN de passage des données en **open-data**.
Actuellement base de données et images aériennes, volonté de diffusion open-data des travaux IA.
- Labellisation existante faite avec 
 - Outils internes
 - Type bureautique SIG (système information géographique) maîtrisés à l'IGN
- Étude/**Etat de l'art sur les outils labellisation IA pour données géomatique**. Diffusée sur site IGN : https://www.ign.fr/publications-de-l-ign/institut/kiosque/etudes/etude_labellisation_ign_2021.pdf



Exemple de données labellisées sur des cartes anciennes. <https://icdar21-mapseg.github.io/tasks/task1/> (©ANR SoDuCo)

Labellisation exemples Gers 2019

- ✓ 1-batiment
- ✓ 2-conifere
- ✓ 3-feuillus
- ✓ 4-ligneux
- ✓ 5-mixte
- ✓ 6-coupe
- ✓ 7-brousaille
- ✓ 8-culture
- ✓ 9-pelouse
- ✓ 10-terre labouree
- ✓ 11-piscine
- ✓ 12-sol_nu
- ✓ 13-surf_eau
- ✓ 14-surf_innondable
- ✓ 15-vigne
- ✓ 16-zone_imper
- ✓ 17-zone_permeable
- ✓ 18-neige



Existant outils /plateforme de labellisation

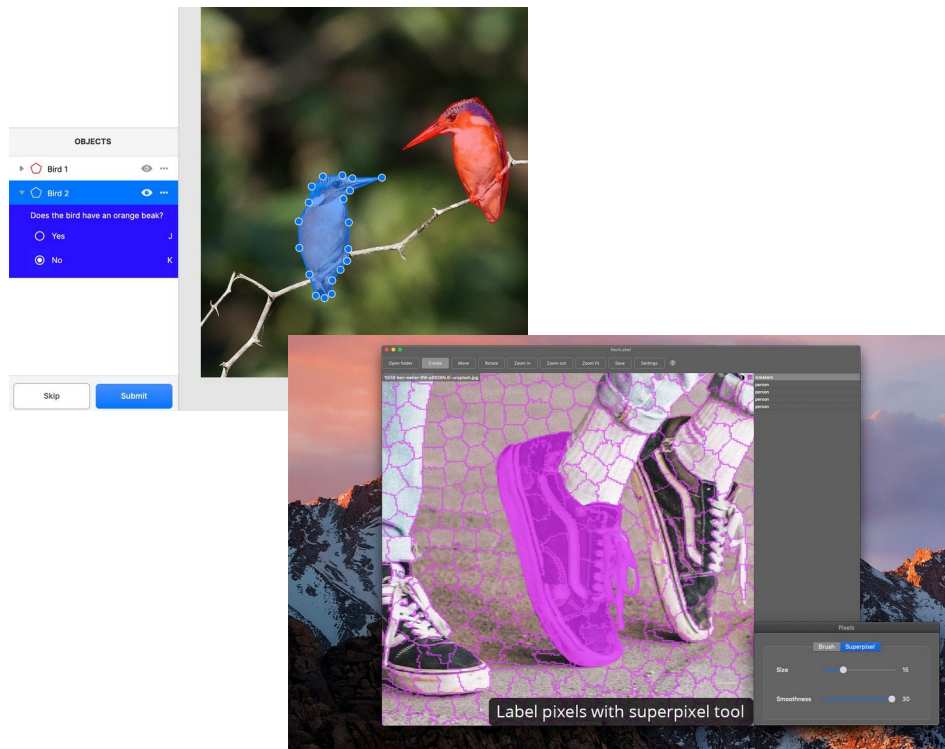
Différents degré d'automatisation :

- **Niveau 0** : annotation **manuelle** pour toutes les tâches
➡ Saisie actuelle **chronophage**
- **Niveau 1** : **annotation assistée (superpixel,**
interpolation vidéo etc., séparation avant/ arrière-plan)

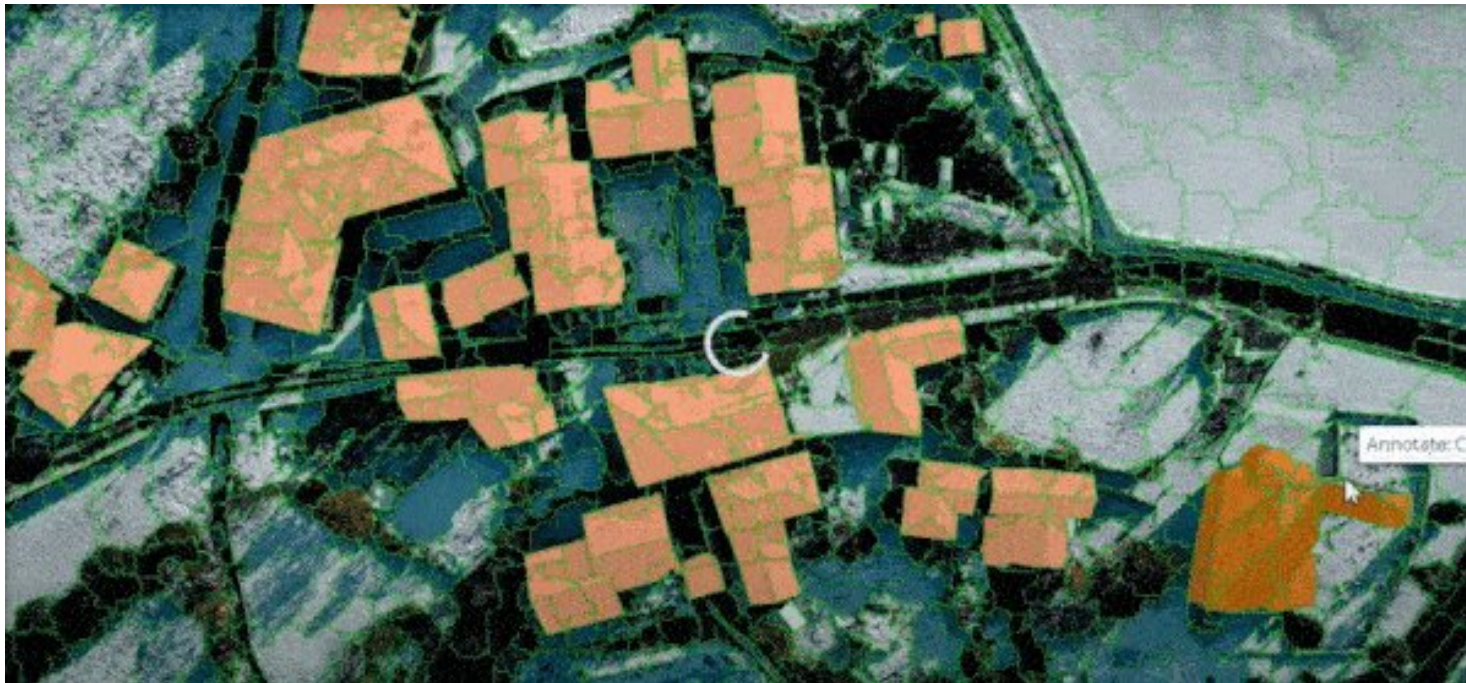
➡ Travaux en cours / **objectif pour les productions actuelles**

- **Niveau 2** : **pré-annotation** (objet/segmentation)
proposition automatique de classe. L'opérateur
vérifie/accepte/corrige les propositions et remplit les
zones qui n'ont pu être pré-remplies.
- **Niveau 3** : pré-annotation quasi-complète. L'opérateur
fait essentiellement du contrôle qualité de l'annotation
automatique et saisie de cas complexes.

➡ Moyens terme / recherche



Ex labellisation via segmentation image

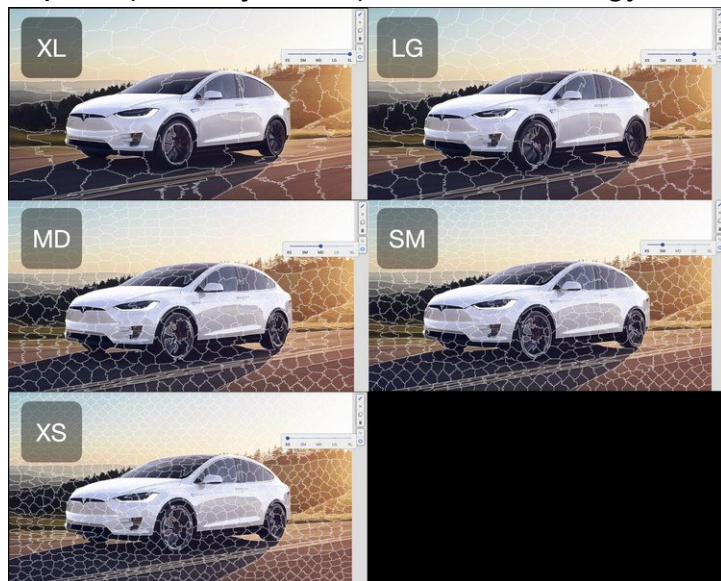


Exemple superAnnotate (<https://www.youtube.com/watch?v=u2Gw6eK94qY>)

Segmentation hiérarchique image

Pour automatisation « niveau de saisie image » → large utilisation de pré-segmentation image. Mais difficile de trouver un paramétrage pour tout objets donc calcul d'une hiérarchie emboîtée de segmentation.
Algo machine learning et non deep avec manipulation graphe (arbres/hierarchies)

Superpixel (slc majoritaire) ex kili-technology



Segmentation hiérarchique

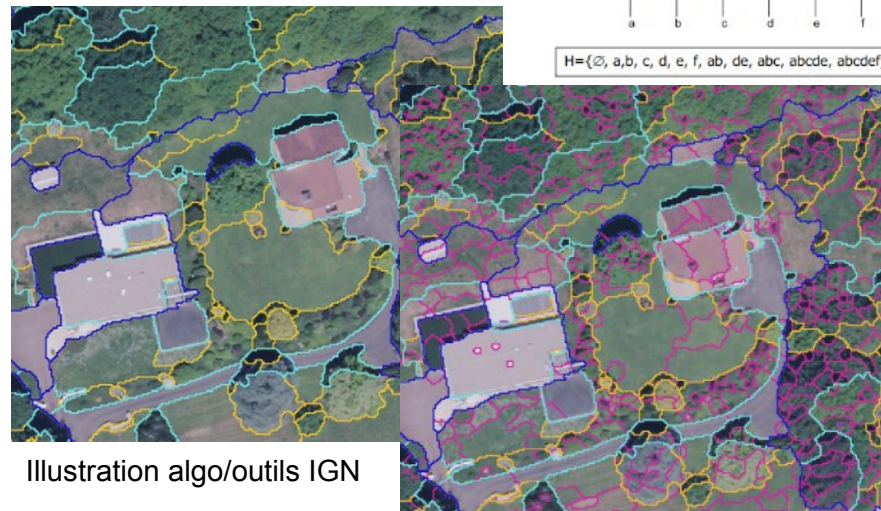
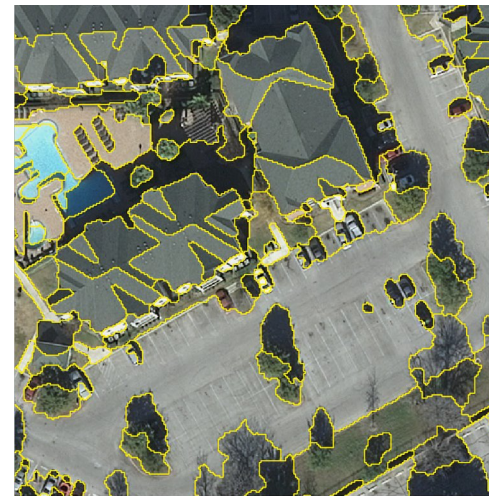
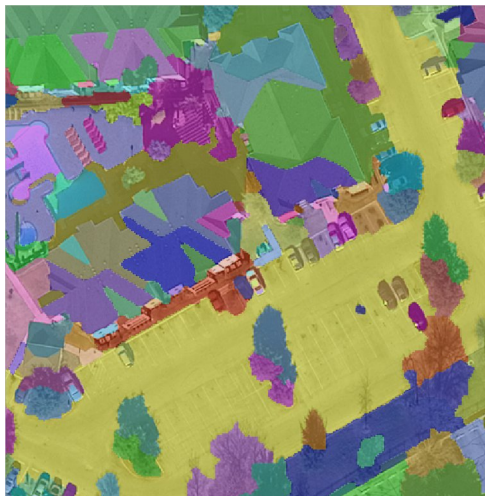
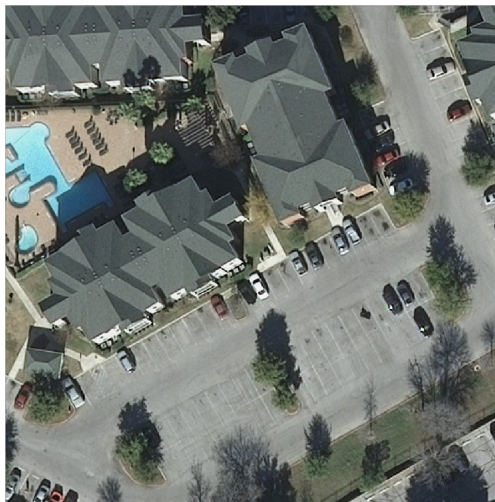
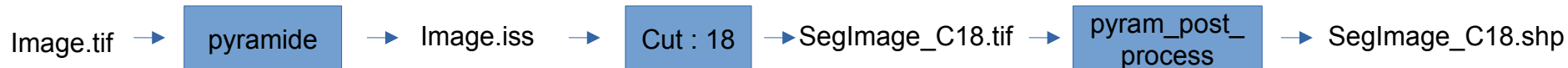


Illustration algo/outils IGN

Code pyram legacy : utilisation



Besoin/commande IGN :

Existant :

- Outils desktop (type plugin)
- Pré-calcul segmentation (code interne C/C++)



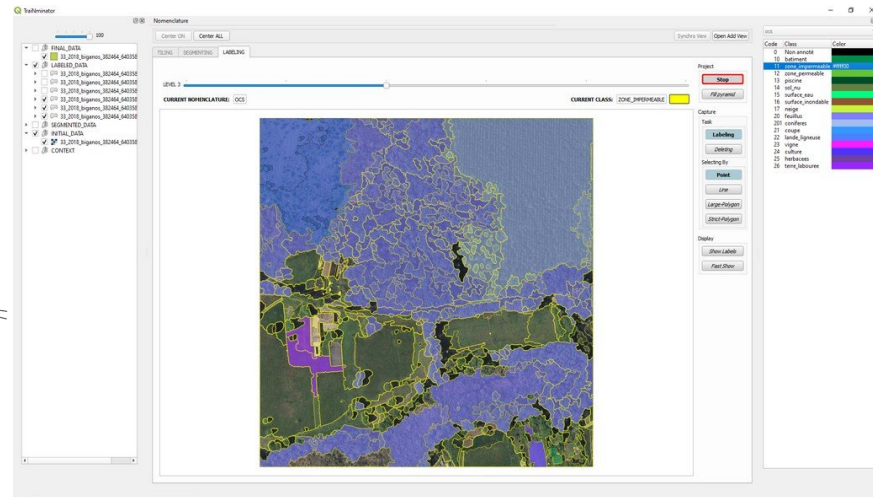
Objectifs Moyen termes :

- Outils pour plateforme client/serveur de labellisation
 - Calcul côté client au max si possible
 - Open-source
 - Test technologies web récentes



Demande Projet :

- Test/reprise code existant rust/wasm de calcul de segmentation
- Intégration du code dans un client pour édition de données
- Gestion données (images et résultats de labellisation) côté serveur.
- (Option) : Plein de chose possible, en particulier gestion de droit/roles ou mode édition de changement (2 vues geo)

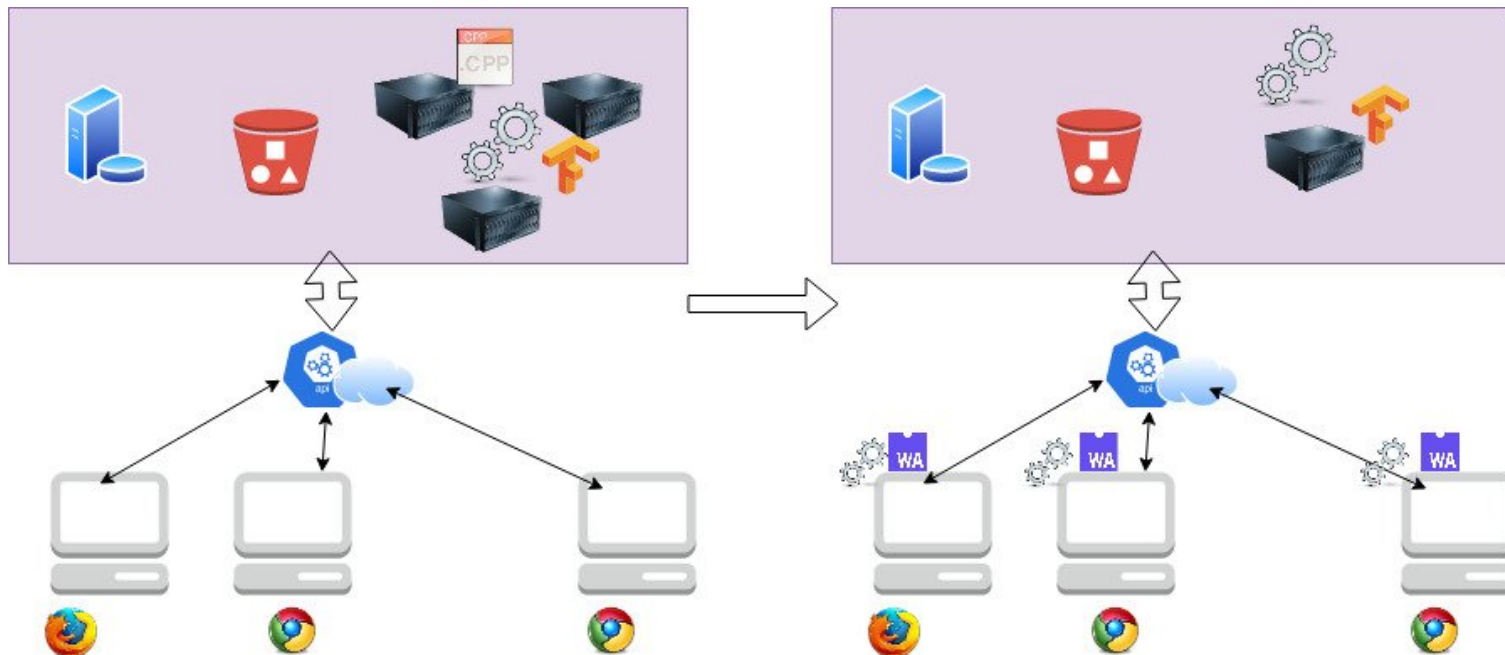


POC d'un outils IGN (plugin SIG qgis) suite stage pour labellisation IA. Stage (été 2020) → reprise (2021)

Fait : Test technologie Rust / Wasm

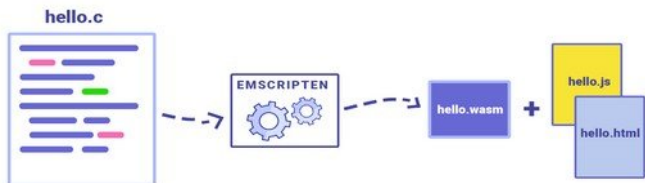


- Poc étudiants EPITA sur refonte code calcul de segmentation. Passage calcul serveur → client possible

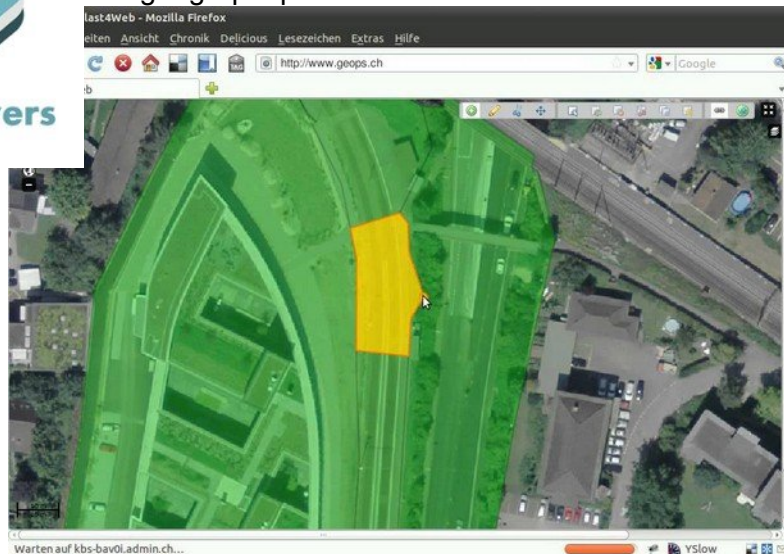


Todo : Intégration librairies js geomatiques

- Intégrer fonctionnalité de saisie sur segmentation dans code javascript (plugin) des librairies open-sources existantes (open layer ou leaflet)

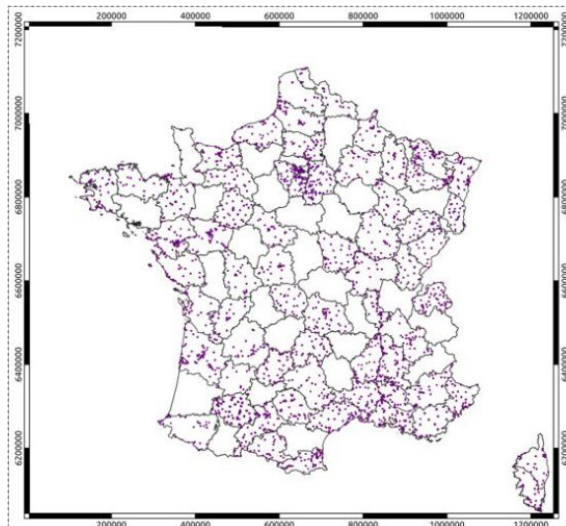


Openlayers : librairie javascript pour visualisation + édition donnés géographiques

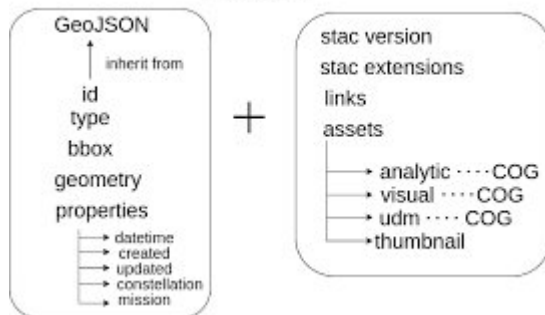


Todo : Gestion données

- Gestion données images (ensemble d'images) : préférence pour techno STAC + COG
- Gestion zones intérêt / patch de saisie / chantier d'annotation ...
- Gestion de nomenclature d'annotation (ensemble de classe)
- Sauvegarde /export des données labellisées



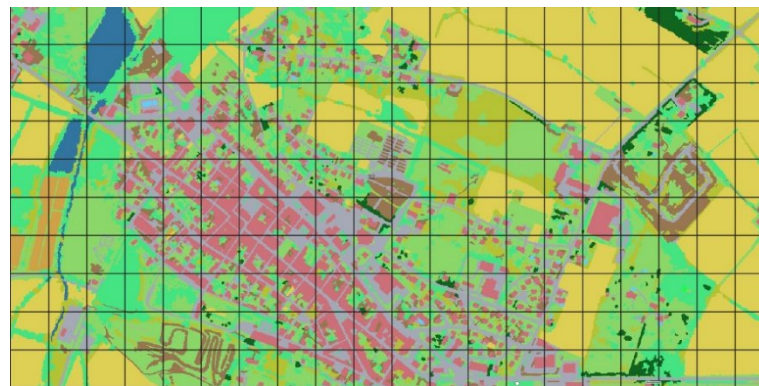
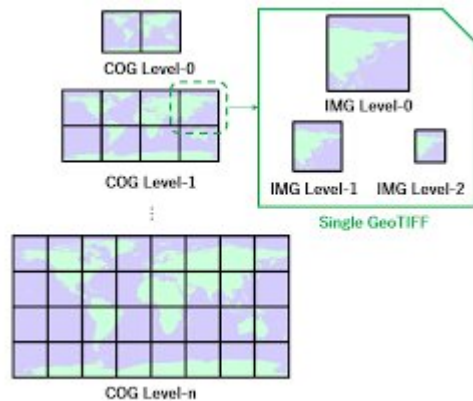
STAC Item





COG

CLOUD OPTIMIZED
GEOTIFF



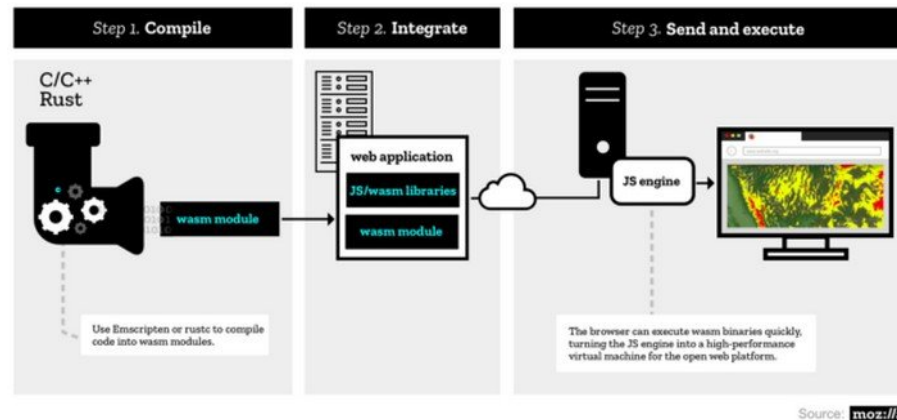
Option : si vraiment vous êtes très rapide...

- Gestion des rôles utilisateurs (gestionnaire de chantier, annotateur, review qualité etc..)
- Gestion d'annotation multi-vue (labellisation de changement entre deux millésimes orthos)
- → pas attendu pour le projet, mais si possible à garder en tête pour avoir un code compatible/évolutif le permettant.

QUESTIONS ? / DISCUSSION

Test technologie Rust / Wasm

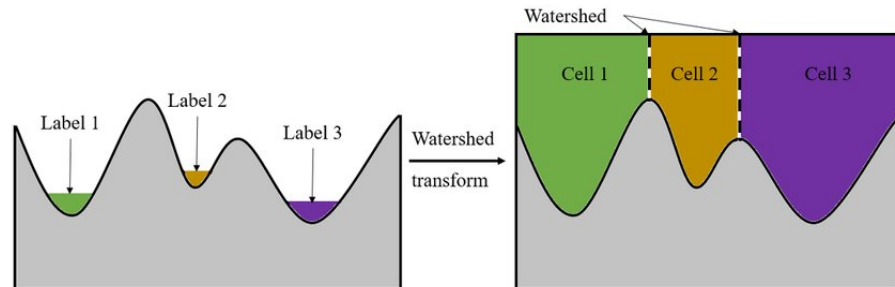
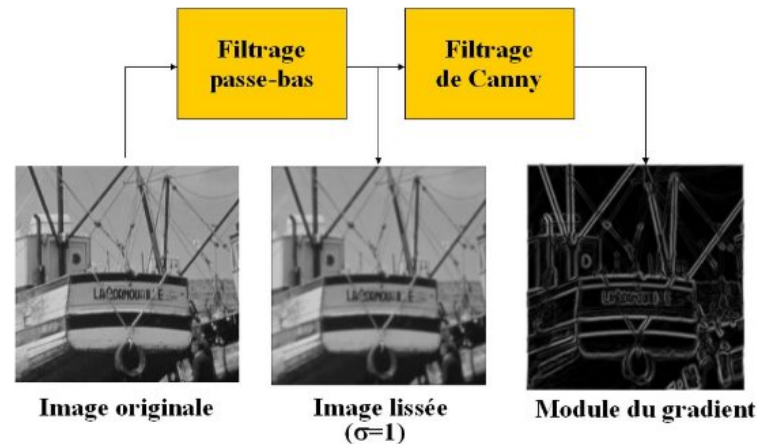
- Intérêt Rust :
 - Alternative C/C++ pour écrire du code « safe » adapté pour code embarqué ou **programmation scientifique bas niveaux** (allocation mémoire image, graphe etc ..)
 - Exécutable ligne de commande (côté serveur) comme code C++
 - **Compilation en WASM pour utilisation côté client** possible et a priori « simple »



- Inspirations pour le projet :
 - Rust : Librairie photon pour traitement d'image côté client à la volé (remplacement imagemagick) <https://silvia-odwyer.github.io/photon/>
 - Rust : Logrocket <https://blog.logrocket.com/webassembly-how-and-why-559b7f96cd71/>
 - Rust : a-b street simulateur pistes cyclables <https://a-b-street.github.io/docs/software/abstreet.html>

Algo Pyram :

1. Chargement image. (ign::BufferImage)
2. Pre processing image :
 1. Sous échantillonnage image
ign::image::underSampling
 2. Ajout Bord ign::image::addBorderMirror
3. Sur-segmentation
 3. Calcul module contour (filtre deriche) par canaux
 4. Aggregation module dériche by Max
 5. Segmentation watershed sur image contours agrégés



Annexes

Institut national de l'information géographique et forestière

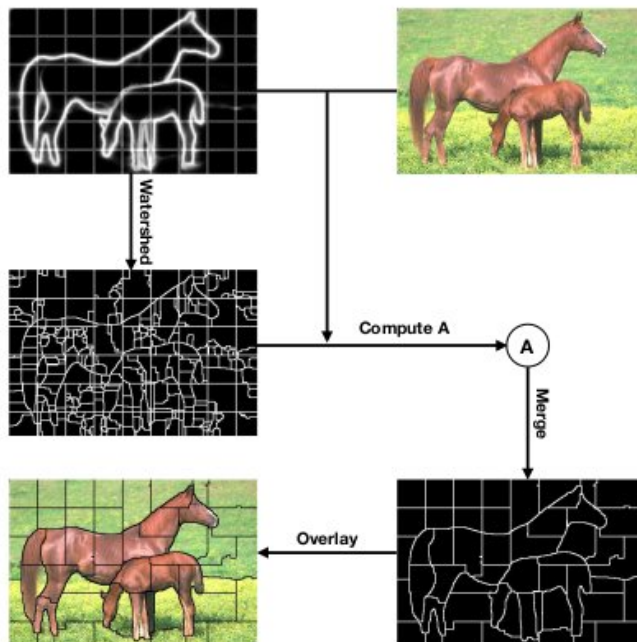
Test labellisation IA OCS 2019.

17

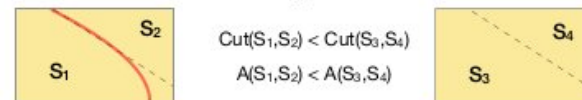
17

Algo Superannotate

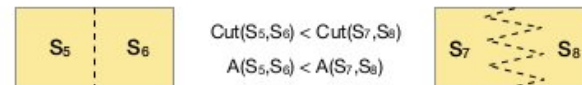
« Fast, Robust and Scalable Clustering Algorithms with Applications in Computer Vision », VAHAN PETROSYAN 2018



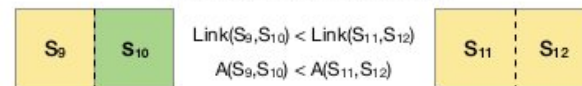
Preserves Object Boundaries



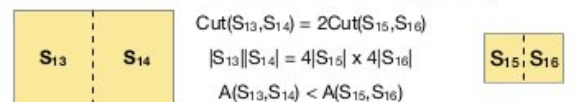
Promotes Smoothness



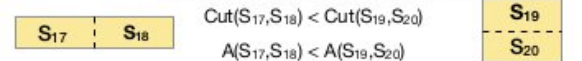
Promotes Similar Color



Promotes Size Homogeneity



Promotes Compactness



Commande projet info EPITA

L'objectif du projet est donc d'essayer de porter un code existant de segmentation hiérarchique d'image de C++ en Rust et de l'utiliser dans le cadre d'application web côté client.

- Tâches à réaliser
 - Revue du code C++ existant (analyse /documentation et mise au propre de l'algorithme).
Fichier code + article mise à disposition par IGN
 - Etat de l'art sur la programmation Rust pour un appel côté client javascript ;
 - Auto-formation sur Rust /Wasm
 - Portage de l'algorithme de segmentation hiérarchique en Rust (sur github)
 - Compilation de ce code en Wasm ;
 - Rédaction des documentations et distribution (packaging) du code. (ex github page)
 - Programmation d'une application JavaScript (exemple simple) permettant de segmenter une image côté client. (Images aériennes test fournies par IGN)
- Livrables/production attendus
 - Une librairie Rust avec un code de segmentation hiérarchique d'image (code + documentation + tests) ;
 - Une méthodologie de compilation de la librairie en WASM et une application JavaScript test