



IGN

**Recherche
& Innovation**

Étude des outils de labellisation - 2021

Etat de l'art des outils existants permettant
d'obtenir et saisir les données d'apprentissage



IGN

INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

Objet: Étude des outils de labellisation pour saisie des données d'apprentissage pour les besoins IGN : recherche et projets.

Apolline DE-WIT -Rédactrice

Nicolas DAVID - Rédacteur

François BECIRSPAHIC - Relecteur

Frank FUCHS - Relecteur

Ce document répond à une demande de la recherche IGN et des projets IA de l'IGN de faire un état de l'art des outils existants permettant d'obtenir et saisir les données nécessaires à l'utilisation de méthodologies de type apprentissage machine supervisé ou semi-supervisé pour des problématiques IGN (essentiellement géomatiques). Cela comprend les travaux actuels sur l'utilisation de l'apprentissage profond dans les processus et recherches IGN.

Le choix a été fait d'avoir une première partie présentant un résumé de l'étude de l'existant et des besoins tandis que les parties suivantes détaillent de façon plus précise différents points choisis de l'étude.

Il est important de souligner que le terme "labellisation" est assez vague et un peu "fourre-tout". Cette étude se concentre principalement sur le sens qui lui est attribué pour les données d'apprentissage profond et dans le cadre du traitement d'images. Une clarification plus détaillée de cette définition est présente dans la première partie.

Concernant les points détaillés : la deuxième partie présente l'existant en outils d'annotation et décrit de façon plus poussée les spécifications fonctionnelles de ceux-ci. Deux autres parties plus techniques suivent sur les méthodologies permettant de rendre plus efficace la labellisation de données images aériennes/géomatiques. La troisième partie aborde les méthodologies de segmentation image existantes et utilisables pour automatiser la saisie de label image au pixel.

Finalement des annexes sont proposées avec des fiches synthétisant la recherche bibliographique faite pour cette étude.

Avertissement : Les comparaisons de fonctionnalités entre solutions de labellisation sont celles existantes au moment de l'étude, i.e le premier trimestre 2021. Les logiciels et plateformes du domaine évoluant très rapidement cette étude n'a donc pas pour objet de conseiller une solution en particulier mais les points importants à considérer si on doit en sélectionner une pour des besoins de labellisation de données géographiques.

De plus cette étude reflète les avis des rédacteurs de l'étude reconnaissent que le choix des critères de sélection contient une part naturelle de subjectivité et peut donc être sujet de débat et discussion.

SOMMAIRE

Préambule : Précision des termes “outils de labellisation” et commande de l’étude.....	3
1 Résumé des besoins IGN et existant en outils de labellisation.....	4
État de l’art en outils et plateforme de labellisation IA.....	4
Besoins et existant IGN en labellisation de données.....	17
Quelques pistes de réponses aux besoins.....	23
2 État de l’art des outils de labellisation pour données géomatiques... 	25
Fonctionnalités générales des outils et plateforme de labellisation.....	26
Fonctionnalités de saisies / interface utilisateur.....	35
Analyse des outils existants.....	43
3 Analyse des méthodologies et outils de segmentation pour labellisation image.....	54
Définitions et enjeux.....	55
Outils de segmentation image et réponses aux enjeux et besoins IGN.....	58
Segmentation multi-échelle.....	69
Comparaison des algorithmes rencontrés.....	75
Conclusion.....	79
Annexe - A : Plateformes et outils de labellisation.....	81
Annexe - B : Fiches fonctionnalités de saisie image.....	85
Annexe - C : Détails outils et algorithmes de segmentation.....	97
Critères d’évaluation.....	114
Bibliographie.....	115

Préambule : Précision des termes “outils de labellisation” et commande de l’étude

Il est important de commencer par préciser ce qui est compris dans le terme “outils de labellisation” et ce qui ne l’est pas dans le cadre IGN et dans cette étude. Ce terme est lié en partie à la commande initiale qui se concentrait sur les besoins des projets IA en cours à l’IGN, en particulier du projet d’automatisation de production de données OCSGE, ainsi que sur les besoins pour les différents projets de recherche du LASTIG (en particulier les équipes STRUDEL / ACTE).

Premièrement la notion de labellisation est très liée, dans le domaine géomatique, au sens qui lui est donné en traitement d’image concernant les données de télédétection mais aussi pour les données plus graphiques comme les cartes (cartes anciennes scannées par exemple). Il s’agit alors des **données qui sont saisies sur ordinateur à partir de visualisation de données images** (photo-interprétation). Il ne désigne donc pas une “labellisation de la qualité”, mais doit être plutôt compris au sens d’annotation ou étiquetage. Les termes annotation/labellisation ou bien *annotate/labeling* en anglais sont d’ailleurs assez interchangeables dans le domaine de l’IA. Dans le cadre des données IGN et géographiques, les données images ou lidar (nuages de points) sont majoritaires dans les données labellisées mais ne représentent pas forcément l’ensemble des besoins.

En télédétection, les données labellisées représentent l’une des sources de données d’apprentissage utilisées par les méthodologies de classifications/segmentation d’image. Il ne s’agit donc pas d’un synonyme pour données d’apprentissages, **en particulier le cas des relevés terrains nécessaires pour certaines études de télédétection n’est pas inclus dans cette étude**. Il s’agit toutefois aussi d’un besoin qui a été exprimé lors d’entretiens menés auprès de différents services de l’IGN, et qui est lié à la mise en place de technologie IA/télédétection.

Les outils de labellisation d’images se sont largement développés pour répondre au besoin croissant de données d’apprentissage pour les méthodologies deep learning. Pour répondre au volume de données à labelliser nécessaire pour les besoins deep learning, le travail est souvent effectué par de multiples personnes. Ainsi, une des notions souvent associée est celle **d’outil collaboratif/distribué en plus de la notion d’interface de saisie** (sur image dans notre cas). Cet aspect a aussi été retenu pour cette étude et en particulier dans le cadre IGN et géomatique il s’agit de préciser quelles sont les différences et similitudes entre les fonctionnalités de l’espace collaboratif IGN ou bien OpenStreetMap et les outils de labellisation.

Enfin la commande sur cette étude concernait principalement les **outils de label/annotation interactifs avec une part importante de saisie manuelle** et non sur tous les outils permettant d’associer des labels à une image par dérivation de base de données externes. En particulier les outils générant de façon automatique des labels pour des images géoréférencées à partir de bases de données géographiques du types BDUi ou OSM sont présentés brièvement mais non détaillés dans la suite de l’étude. On considère en effet ces outils comme des outils SIG plutôt standards en termes de fonctionnalités (extraction, sélection de données, rastérisation de données vectorielles).

1 Résumé des besoins IGN et existant en outils de labellisation

Nous présentons ici un résumé des éléments constatés lors de l'étude en essayant de faire ressortir les points de compréhension importants ainsi que les points saillants pour la suite des travaux IGN sur ce sujet.

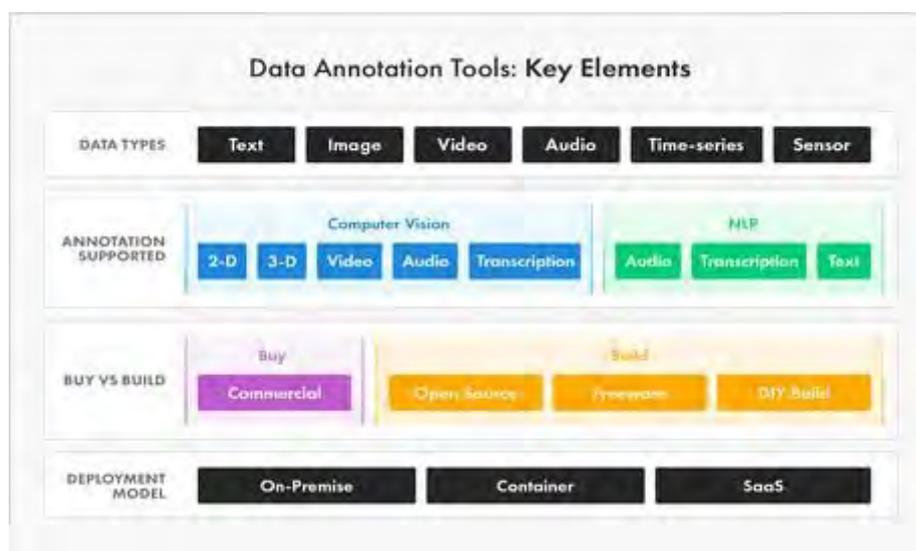
État de l'art en outils et plateforme de labellisation IA

L'émergence des techniques d'apprentissage profond a entraîné une évolution et apparition de nombreux outils et plateformes pour répondre aux besoins de saisie de données d'apprentissage. Les larges jeux de données nécessaires à ces méthodologies d'apprentissage supervisé ayant conduit à un besoin d'outils dédiés pour diminuer leur coût de constitution. Afin de pouvoir décider si ces outils peuvent répondre à un besoin de labellisation spécifique il faut d'abord commencer par en définir les principales fonctionnalités et critères.

Fonctionnalités générales

Une première typologie haut niveau des outils peut être faite en fonction des deux critères suivants :

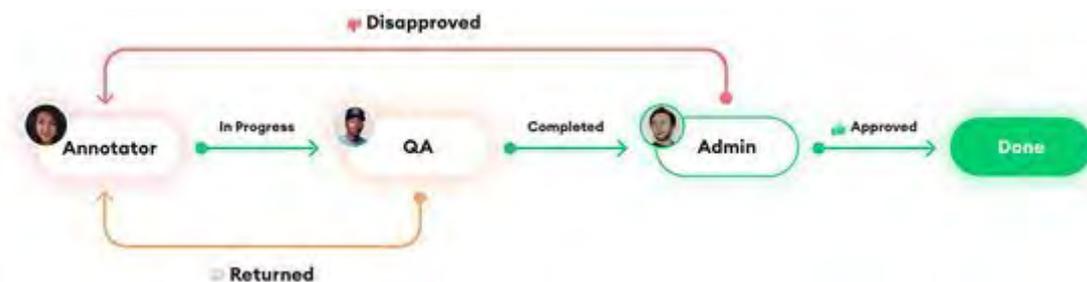
- **Plateforme vs outil/logiciel** : Parle-t-on seulement d'un logiciel de saisie image fonctionnant sur un ordinateur personnel ou bien d'une application ou service web permettant de travailler à plusieurs utilisateurs et offrant en plus des fonctionnalités supplémentaires à la saisie proprement dite des labels. Dans le cas des plateformes le mode de déploiement (seulement cloud ou aussi sur infra propre) est à prendre en compte.
- **Propriétaire vs open-source** : Critère important s'il y a un besoin de reproductibilité et de diffusion d'outils/savoir. Mais rentrant aussi en compte sur les questions de coût des outils de labellisation ou bien sur les possibilités d'adapter/optimiser l'outil de labellisation à un besoin spécifique ou de l'héberger sur une infrastructure propre pour des raisons de sécurité.



Critères importants d'analyse des plateforme & outils de labellisation ([cloudfactory](#))

Ensuite les fonctionnalités importantes pour l'adéquation d'un outil à un besoin de labellisation spécifique sont principalement :

- **Type de donnée à labelliser** : image, texte, nuages de points, vidéo, audio etc.
- **Tâches de labellisation disponibles** : Pour les données géomatiques et les images/nuages de points cela comprend en particulier la distinction entre saisie de tag/classification (une catégorie par image), saisie de boîte englobante/objet, saisie de segmentation image (une catégorie par pixel) ou bien d'instance (une catégorie et identifiant par pixel).
- **Gestion de la qualité** : Outils et processus mis en place pour garantir la qualité des annotations. Généralement implémenté par une étape de contrôle (*review*) des données labellisées et par des processus de saisie multiples sur certaines parties du jeu de données.



SuperAnnotate : exemple de processus qualité pour la labellisation

- **Suivi et gestion de production** : Fonctionnalités allant de la gestion des données en entrée et création de projet, à la répartition des tâches de saisies et contrôles qualités entre de multiples personnes et au suivi de l'avancement par projet.
- **Sécurité** : respect des contraintes de type RGPD, protection de l'infrastructure et des échanges/transmission de données (clef privée/publique, méthode d'authentification), fonctionnalité d'identification et de gestion de rôles utilisateur.

L'étude de ces différents critères et fonctionnalités permet déjà de faire un premier filtre efficace des outils et plateformes existants en fonction des besoins et contraintes exprimés pour un projet de labellisation de données spécifique. Dans le cas des projets IGN d'automatisation de la production OCSGE et BDforêt par exemple, cela supprime tous les outils ne supportant pas la labellisation de données images, puis parmi ceux ayant une offre de labellisation image, on peut filtrer ceux ne supportant pas les tâches de labellisation de type segmentation sémantique.

REM : Ces différents aspects sont détaillés de façon plus précise dans la deuxième partie de l'étude.

Type de labellisation image

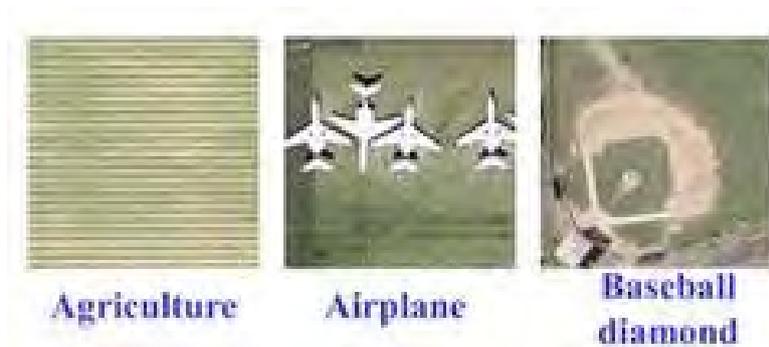
Le type de tâche de labellisation possible avec une plateforme d'annotation est un des premiers critères à regarder car l'outil est inutile s'il ne permet pas d'effectuer la tâche correspondant aux besoins en données d'apprentissage.

En ce qui concerne les données de type image (et nuage de points) sur lesquelles portent principalement cette étude il faut distinguer quatre types de tâche de labellisation :

- Classification d'image
- Détection d'objet
- Segmentation sémantique d'image
- Segmentation d'instance

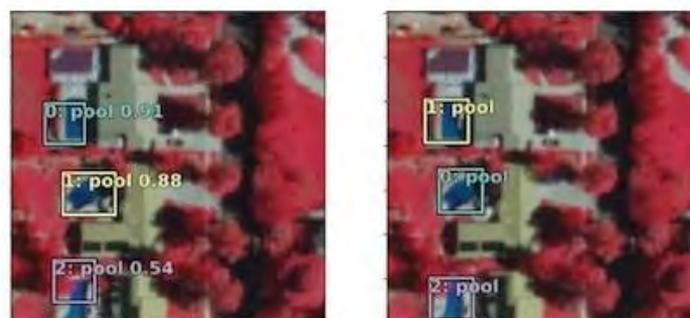
On détaille ici les tâches dans le cadre de l'image mais la distinction est assez similaire pour des données 2D ou de type nuage de points.

Classification d'image : affecte un ou plusieurs labels/tag à une image. Cela est aussi appelé classification de scène en télédétection. Une particularité de la géomatique par rapport aux cas plus standard en traitement d'images est que les images de types ortho sont, d'une part, très larges (jusqu'à une couverture mondiale) et, d'autre part, les vignettes/tuiles qui peuvent en être extraites ne sont pas centrées sur un objet d'intérêt particulier mais sur une coordonnée géographique plus ou moins choisie. Donc dans certains cas en géomatique la notion de label par tuile/imagette n'a pas forcément de sens.



Exemple de classification d'image en télédétection : [UC merced land use dataset](#)

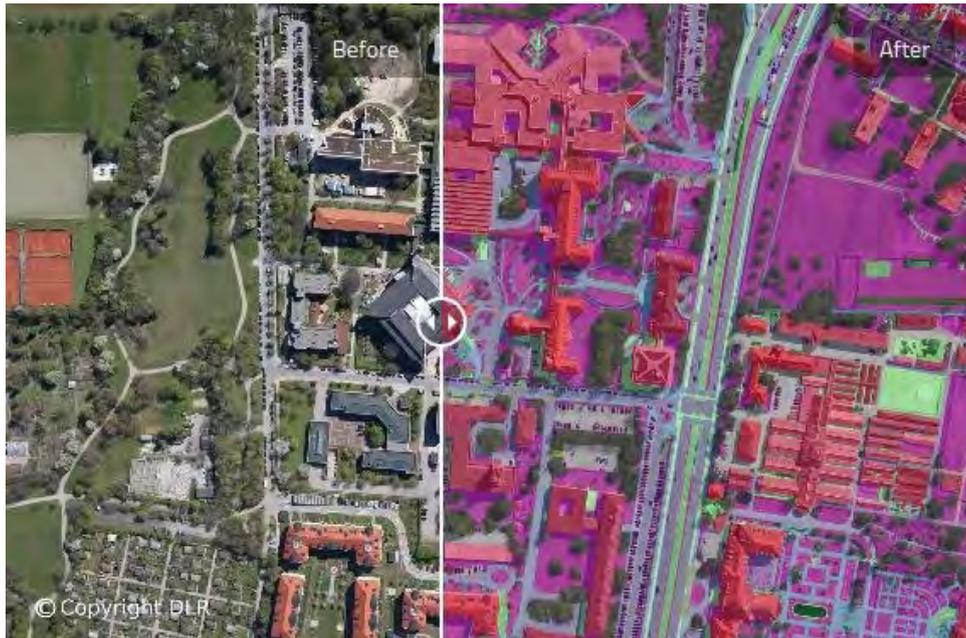
Détection d'objet : Associe une boîte englobante à un objet d'intérêt. En géomatique cela est surtout utile pour les objets individuels de type bâtiments/piscines ou bien arbres isolés. Il s'agit dans la majorité des cas de boîtes englobantes non orientées.



Exemple de détection objet sur des piscines ([arcgis pro](#), ESRI©) :

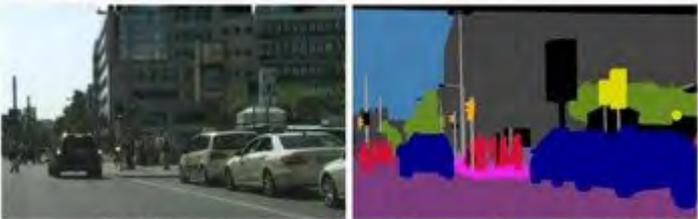
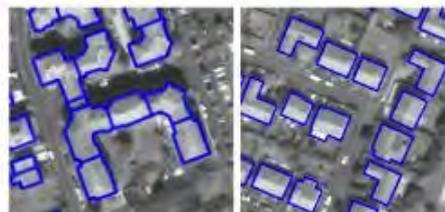
Segmentation sémantique et Segmentation d'instance : Associe des valeurs à chaque pixel d'une image. Pour la segmentation sémantique il s'agit d'une valeur

encodant la classe à laquelle appartient le pixel. Il s'agit de la tâche la plus courante actuellement pour les données de type occupation du sol.



Exemple de données labellisées pour la segmentation sémantique (image [ternowAI/DLR](#)©)

Dans le cadre de la segmentation par instance il s'agit d'avoir une valeur différente de pixel pour chaque objet. L'objet peut ensuite être associé à une classe via son identifiant. Enfin dans le cadre de la segmentation panoptique chaque pixel a un identifiant et une classe d'associée. La différence entre instance et panoptique vient de la différence vue entre "thing" et "stuff", soit entre objet et truc/matière. Les objets d'une même classe ont un identifiant qui permet de les distinguer alors que les "choses/trucs" comme le ciel ou la route n'ont pas d'identifiant associé et sont considérés comme uniques et continus.

Exemple de segmentation et instance pour données images terrestre (source " panoptic segmentation ")	Différence segmentation sémantique/instance pour bâtiment
 <p>(a) image (b) semantic segmentation</p>	
 <p>(c) instance segmentation</p>  <p>(d) panoptic segmentation</p>	

Comparaison outils SIG et outils de labellisation image IA

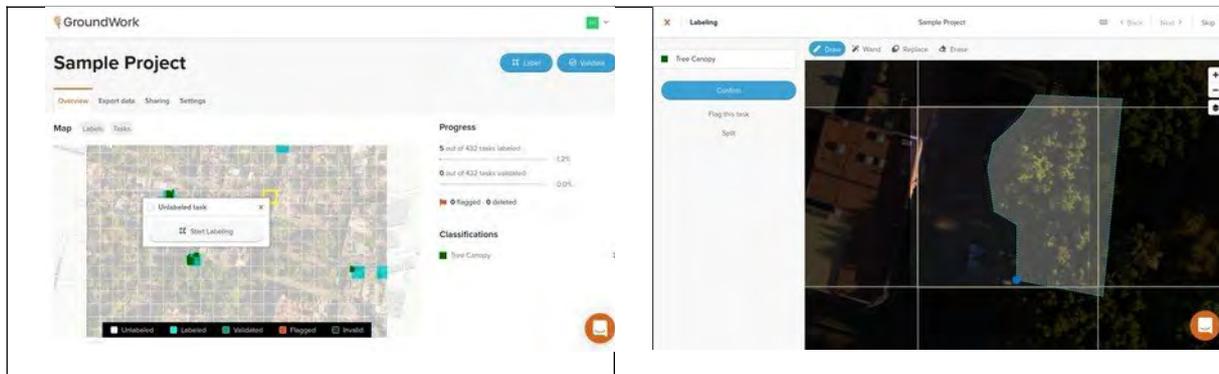
Les tâches de saisie de label sur images dans le cas de segmentation sémantique ou d'instance sont assez proches de la saisie d'éléments topographiques ou cartographiques faite avec des outils SIG classiques. Il semble donc légitime, aussi bien pour les besoins de labellisation en géomatique qu'en télédétection, de **distinguer les outils SIG offrant des fonctionnalités de labellisation IA des outils IA de labellisation de données images compatibles avec les données géographiques**. En effet, les logiciels SIG étant déjà utilisés à l'IGN, si l'un d'eux peut répondre aux besoins de labellisation, cela faciliterait la prise en main de l'outil par les opérateurs et son intégration dans les processus existants, plutôt que de former les personnes à un nouveau type de logiciel. Il y a toutefois quelques différences notables entre les deux types de saisies, données cartographiques vs données d'apprentissage qui se retrouvent dans les fonctionnalités présentes dans les deux catégories de logiciel :

- Une saisie plus précise (proche du pixel) pour les données d'apprentissage par rapport à une saisie d'éléments pour des bases de données géographiques.
- Des fonctionnalités d'affichages plus complètes pour les outils SIG et adaptées aux images aériennes et spatiales (étendues larges, multiples canaux spectraux ou multiple date/source de données). Alors que les outils de labellisation fonctionnent principalement sur de petites images en RVB.
- Des fonctionnalités plus intégrées de gestion de production/contrôles qualité dans le cadre des outils de labellisation IA.

Outils existants ayant des fonctionnalités de saisie SIG et de labellisation IA :

On peut noter qu'il existe aujourd'hui **peu d'outils ou de plateformes de labellisation de données qui offrent un support poussé pour les données images utilisées en télédétection ou cartographie**. Parmi les outils et plateformes ayant de telles fonctionnalités ou ceux ciblant spécifiquement des cas géomatiques/télédétection on peut citer :

- La plateforme [groundwork](#) de la société Azavea. Solution de labellisation issue de l'écosystème des outils géomatiques mais dont les fonctionnalités sont relativement basiques par rapport aux solutions de labellisation et avec donc un temps de saisie relativement important. Par rapport aux outils SIG classiques, son apport est essentiellement le découpage de la labellisation en tâches élémentaires qui peuvent être distribuées sur plusieurs personnes.



Exemple d'interface de saisie de l'outil [GroundWork](#) (Azavea ©)

- Le logiciel Arcgis Pro (Esri©) propose également quelques [fonctionnalités de saisie](#) plus orientées sur les données d'apprentissage images, via une saisie sur segmentation image ou par polygone et boîte englobante. Le logiciel [Ecognition](#) (Trimble©), orienté télédétection peut aussi être utilisé pour la saisie de données d'apprentissage via la constitution de segmentation/classification d'image. Ces outils offrent un bon support des fonctionnalités de visualisation de données géographiques mais ne possèdent pas de fonctionnalités de gestion de production ou qualité et garde des interfaces de saisies relativement peu optimisées pour la saisie de données d'apprentissage.
- Certains outils ou plateformes de labellisation génériques montrent des exemples avec des images aériennes, souvent en RVB, qui illustrent la compatibilité des méthodologies employées avec les données géomatiques. Toutefois l'utilisateur doit alors effectuer des pré-traitements/post-traitements de données sur les entrées/sorties de l'outil de labellisation pour le rendre compatible avec les données géomatiques (découpage des images, récupération du géoréférencement après annotation, etc.). C'est le cas par exemple de l'outil de [superannote](#).

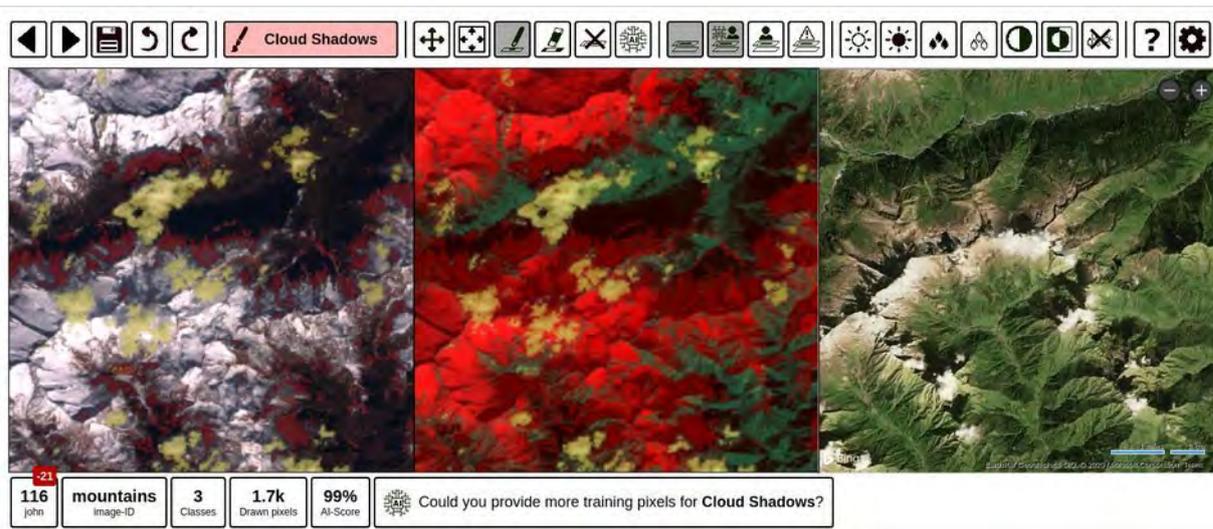


Exemple de saisie bâtiments, routes et végétation haute avec l'outil [superannote](#)©

- La plateforme de labellisation IA Labelbox offre des fonctionnalités compatibles avec [les données géomatiques](#), en particulier un support des données de type [slippy map \(XYZ\)](#) et une gestion des données multispectrales (multilayer). Cela facilite un peu l'utilisation des données

géomatiques mais reste limité en termes de fonctionnalités de visualisation.

- Il existe aussi des outils développés pour un besoin particulier (constitution de jeux d'apprentissage). Par exemple l'outil [IRIS](#) conçu pour la labellisation de nuages dans les données Sentinel-2 ou bien la plateforme open-source [AIDE](#) produite pour la détection d'animaux à partir d'images aériennes. Ces interfaces sont alors relativement bien adaptées à un besoin particulier mais ne sont pas facilement adaptables ou n'ont pas un niveau de maturité permettant de les déployer facilement.



Exemple d'interface de saisie de l'outil IRIS, image sentinel-2 pour la labellisation de nuages (IRIS, ESA-PhiLab©)

Cas des outils de labellisation automatique de données géographiques :

Une raison pouvant expliquer le faible nombre d'outils de saisie de labels pour les données géomatiques est le fait que de par la nature spatiale des données, **une base de données géographiques peut être appariée facilement à une donnée image géoréférencée**. Ainsi on peut constater qu'il existe un certain nombre d'outils permettant de générer des labels compatibles avec les besoins des bibliothèques deep learning à partir de base de données géographiques (OSM par exemple) et des images aériennes ou satellites. Ces outils offrent des fonctionnalités de sélection et filtrage des BDD géographiques, de rasterisation de ces données et de formatage dans des formats compatibles avec les frameworks IA.



Résultat de labellisation "automatique" par [ohsome2label](#)

On peut citer une liste non exhaustive de ces outils : [Label-maker](#) par la société “developpement seed” qui utilise les données OSM en vérité terrain, [ohsome2label](#) qui utilise aussi les données OSM mais permet de plus de gérer l'historique des données pour avoir une donnée extraite plus cohérente avec la date d'acquisition de l'image, la librairie raster-vision qui propose dans sa chaîne de traitement une étape de génération de label (étape “[chip](#)”) à partir de la plupart des sources de données vecteur et images possible (via la librairie gdal) , ou bien encore les outils [Geolabel-maker](#) de “makina corpus” et [Buzzard](#) d’“Earthcube” et le plugin Arcgis [LabelRS](#).

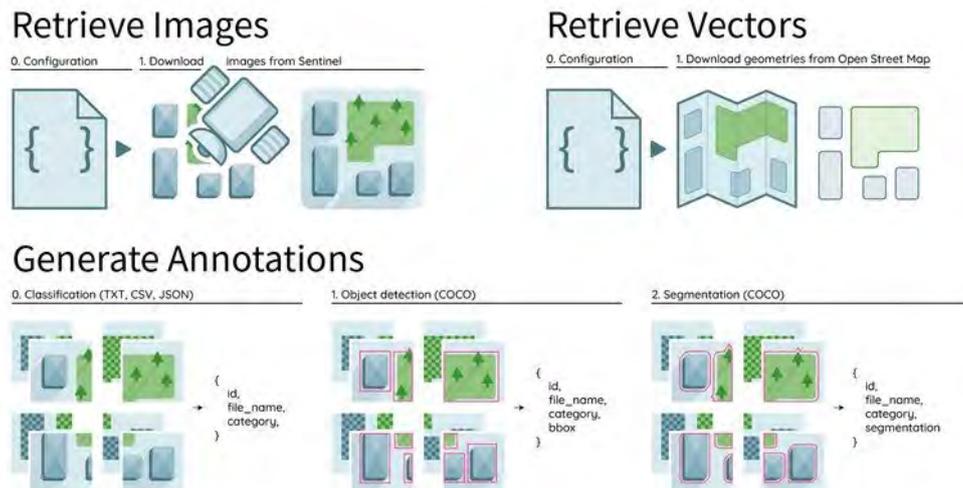
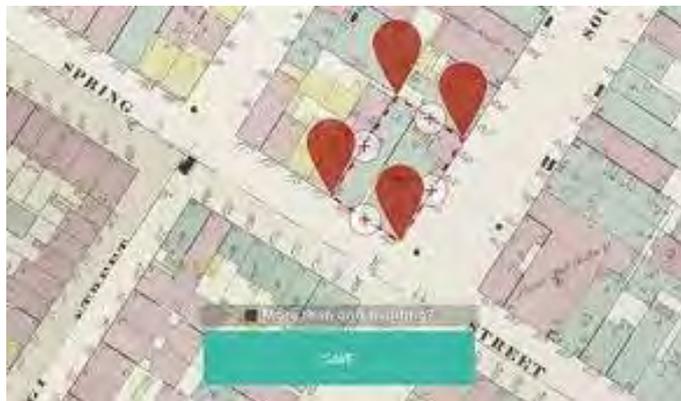


Illustration de l'annotation/labellisation automatiques à partir de BDD géographique et image géoréférencées (©makina-copus).

Utilisation des outils d'éditations collaboratives des SIG pour la labellisation IA :

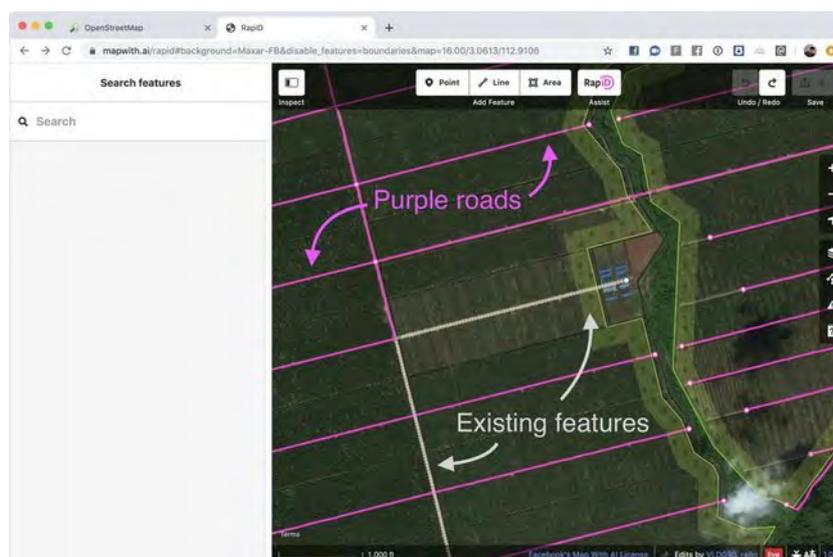
Les plateformes de labellisation IA offrent souvent des fonctionnalités d'édition collaborative de type répartition de rôle et distribution de tâches. La notion d'édition collaborative (crowdsourcing) est également bien connue dans le domaine géomatique, cependant **l'écosystème d'outils d'édition collaborative SIG semble peu utilisé pour la constitution de données d'apprentissage en télédétection**. Les exceptions notables sont l'utilisation et l'adaptation des outils OSM pour le cas de la vectorisation de cartes anciennes et l'existence d'outils de crowdsourcing pour l'édition et le contrôle qualité de données géographiques qui ont des similarités avec les outils et plateformes de labellisation IA pour les données images.

Concernant le cadre de la vectorisation de cartes anciennes, un des outils précurseurs est “[building inspector](#)” proposé par la New York Public Library. Ce projet offre un ensemble d'outils collaboratifs et interactifs permettant de géoréférencer puis vectoriser des cartes anciennes sur New-York. En France, on peut noter l'initiative de geohistoricaldata ([article](#)) sur les cartes anciennes de Paris ou de Cassini. Plus récemment, Google a repris des briques des outils OSM et du projet Building Inspector pour son projet [recreating historical streetscapes](#) (Karrta labs).



Saisie de bâtiments via building inspector (©new york public library)

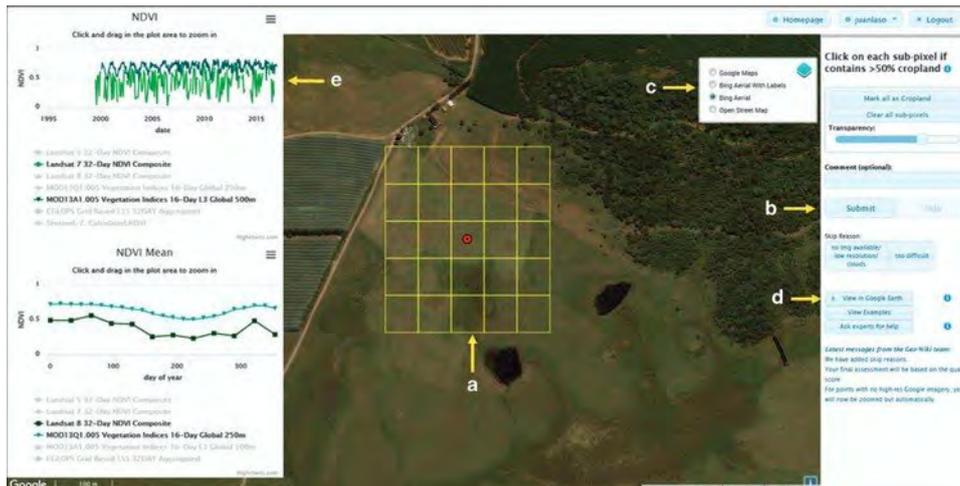
Par rapport aux outils de labellisation IA, ces outils de labellisation plus cartographiques ne répartissent pas forcément la charge de travail en tâches de saisies élémentaires, l'édition pouvant être faite en "temp réel" sur de larges zones et les contributeurs se répartissant librement les zones de saisie. D'autre part, **les interfaces de saisie sont relativement simples et avec peu d'automatisation ou de recherche d'efficacité** (nombre de clic) mais par contre elles sont bien adaptées à une tâche spécifique, comme par exemple l'appariement/géoréférencement de carte. Un exemple notable d'adaptation de ces outils avec des techniques d'aides à l'annotation est le projet [mapwith.ai](#) de Facebook qui propose un éditeur "RapidID" (dérivé de l'éditeur Id de l'écosystème OSM) avec une aide à l'édition des routes.



Proposition de tracé de route par Mapwith.ai (©Facebook)

Une autre famille d'outils collaboratifs géomatiques ayant des similarités avec les outils de labellisation IA est celle des applications de type [geo-wiki](#) utilisées notamment dans le projet [LandSense](#). L'objectif est alors similaire à celui de la classification d'image et consiste à déterminer une classe ou valider une classification d'image sur un ensemble d'échantillons (vignettes images). La validation peut soit être faite sur le terrain via des applications mobiles ou bien en ligne à partir de photo-interprétation d'image. Cette gestion de données terrain en fait plus un outil de gestion de données d'apprentissage (ou validation)

que de labellisation. Les images servant pour la validation pouvant être de plus haute résolution spatiale que celle utilisée lors de la classification.



Saisie/validation d'échantillons dans geo-wiki (IIASA©)

Bilan comparaison outils SIG / labellisation IA :

On peut donc constater qu'aujourd'hui il existe seulement un faible recouvrement entre les technologies des outils SIG "standards" et les outils et plateforme de labellisation IA.

En particulier, **les plateformes de labellisation IA offrent des solutions plus complètes en termes de production de données** en intégrant nativement des fonctionnalités de contrôles qualités et gestion de production/distribution des tâches qui ne sont pas présentes dans les outils SIG. De plus, **les outils de labellisation IA offrent souvent des interfaces utilisateurs de saisies plus évoluées et efficaces** avec divers niveaux d'automatisation.

Intégration et automatisation de la saisie de données d'apprentissage

Bien que les premiers outils de labellisation aient été relativement simples, le coût associé au volume des données à labelliser a conduit rapidement à une recherche d'automatisation maximum des tâches de saisies.

On peut alors distinguer, selon les outils et plateformes, plusieurs niveaux d'automatisation possibles.

- Niveau 0 : annotation manuelle pour toutes les tâches
- Niveau 1 : annotation assistée (superpixel, interpolation vidéo etc., séparation avant/ arrière-plan)
- 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 préremplies.
- Niveau 3 : pré-annotation quasi-complète. L'opérateur fait essentiellement du contrôle qualité de l'annotation automatique et de la saisie de cas complexes.

Pour illustrer le type d'automatisation possible au niveau 1, on peut regarder l'évolution des méthodologies d'extraction d'objet (ou extraction avant/arrière-plan) à partir de marqueurs (points, boîte englobante, traits...). Cela est utile pour

la segmentation sémantique d'objet et permet de diminuer le temps de saisie en nombre de clics/actions nécessaires pour obtenir un contour fin d'un objet.

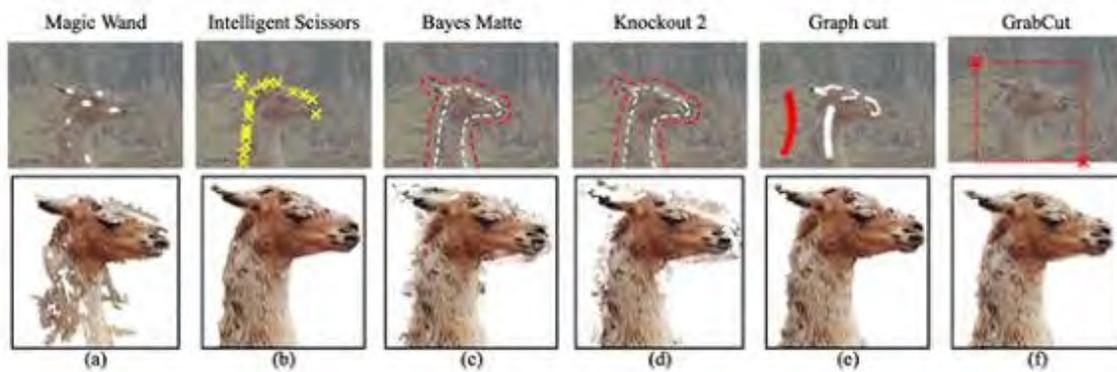


Illustration de l'évolution des techniques d'extraction automatique de contours (ordre chronologique). Ligne haut élément saisie par l'opérateur, ligne bas résultat d'extraction d'objet correspondant (© C Rother, article [Grabcut](#))

Le niveau 1 d'automatisation correspond souvent à remplacer la tâche de saisie manuelle par une tâche d'extraction d'objet guidée par l'utilisateur (via des marqueurs) et est donc plus simple, d'un point de vue algorithmique, que la tâche d'extraction sans guides. Cette dernière étant souvent l'objet de la création du jeu d'apprentissage.

Il faut noter que certaines de ces méthodes de saisie demandent, pour être efficaces, que l'utilisateur comprennent bien les marqueurs à fournir à l'algorithme : i.e ce qui représente de bons exemples et contre-exemples. Elles ne sont donc efficaces que pour des utilisateurs expérimentés et faisant beaucoup de saisie, mais peuvent être contre-productives pour un utilisateur occasionnel et/ou ne connaissant pas l'algorithme utilisé.

Ici les niveaux 2 & 3 utilisent alors **des méthodologies de type apprentissage actif (active learning) et il y a une boucle entre les données labellisées et l'apprentissage d'un modèle deep learning** qui sert à prédire les labels (annotations).



Boucle de labellisation avec pré-annotation apprises (© [labelbox](#))

Les différents niveaux d'automatisation ne sont pas forcément indépendants car il peut être nécessaire de d'abord utiliser des outils de niveau 0 ou 1 avant de pouvoir utiliser des outils de niveau 2 ou 3 et amorcer l'apprentissage d'un premier modèle. En effet ceux-ci ont besoin d'un modèle capable de prédire les labels de façon imparfaite mais quand même assez fiable pour apporter un gain par rapport à une saisie à partir de zéro. Il faut alors soit partir sur un modèle entraîné à partir de quelques données déjà labellisées par un autre moyen

(niveau d'automatisation 0 ou 1 par exemple) soit partir d'un modèle entraîné sur une tâche similaire et l'adapter au fur et à mesure de la saisie des nouvelles données.

Il faut aussi noter que l'automatisation de niveau 2 ou 3 impose souvent une architecture d'outils/plateformes plus complexe du fait de la boucle d'entraînement. Il est alors souvent nécessaire d'avoir une architecture de type client-serveur avec la saisie de labellisation faite côté client et l'entraînement de modèle et la prédiction faite côté serveur avec des ressources de calculs pouvant être conséquentes.

Enfin au niveau de la création de données géographiques il y a aussi de fortes similitudes entre les méthodologies d'automatisation des données d'apprentissage ou les méthodologies pouvant être mises en place pour automatiser la production/saisie de données géographique directement donc on peut se poser la question entre l'automatisation de la saisie de données d'apprentissage ou l'automatisation de la production des données géographiques elle-même.

Outils et briques logiciels recherche et open-source

Le besoin en outils d'annotation a été d'abord constaté par les équipes de recherches en apprentissage profond afin de conduire leurs expérimentations. De par les objectifs de reproductibilité et de sciences ouvertes assez fort dans cette communauté, plusieurs outils de labellisation open-source ont alors été élaborés. Cette ouverture d'outils de labellisation a aussi été continuée par certaines communautés d'intérêt ou par de grandes entreprises comme Microsoft (outil [VoTT](#)).

Au niveau de la typologie de ces outils open-source on peut distinguer :

- Les outils développés par une communauté autour d'une thématique précise afin de répondre à ses besoins de manière optimale. Par exemple [ImageTagger](#) dans le cadre des compétitions de robotiques (RoboCup)

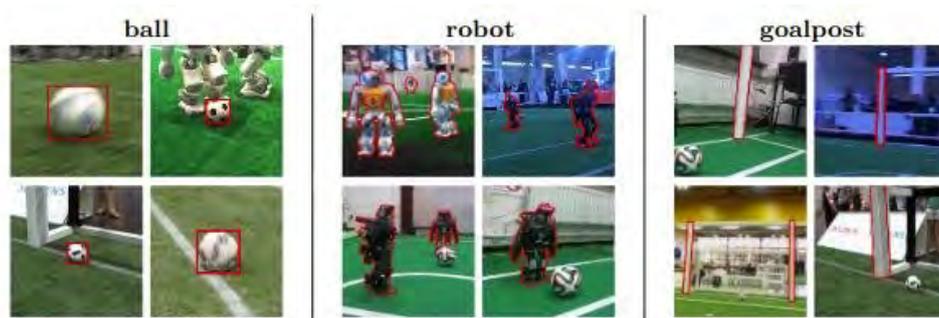


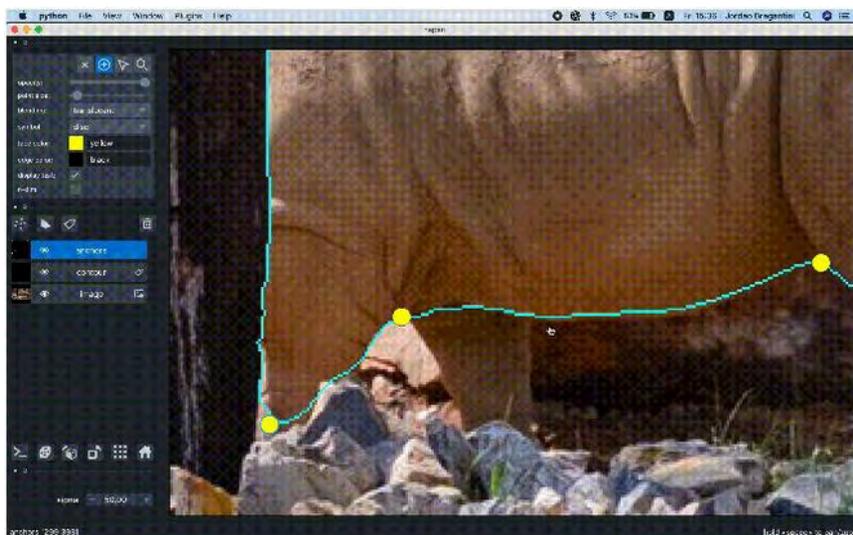
Fig. 1: Exemplary labels from categories common in RoboCup soccer. Precise labels are created to allow learning of exact object localization.

(© Fiedler, *ImageTagger: An Open Source Online Platform for Collaborative Image Labeling*)

- Les outils généralistes de labellisation pour la saisie de jeux d'apprentissage. Ceux-ci peuvent être très simples avec peu de fonctionnalités comme [LabelImg](#) qui permet juste la saisie de tags image ou de boîtes englobantes sur une liste de fichiers images. Ou bien ils

peuvent avoir des fonctionnalités de saisie plus complètes comme django-labeler qui possède des outils d'édition plus automatiques pour la segmentation sémantique, comme une implémentation de l'algorithme Dextr permettant de segmenter un objet à partir de la saisie de 4 points extrémités.

- Les outils associés à un algorithme ou une méthodologie spécifique, en particulier dans les recherches autour de l'automatisation de la labellisation (grabber, pixel-annotation tool, livewire-gui). Ce sont souvent des outils de démonstration associés à un article.



Exemple d'outils pour tester une méthodologie de saisie (© Grabber: A Tool to Improve Convergence in Interactive Image Segmentation). Image de début de saisie, avant ajout de la jambe manquante.

Concernant les stacks techniques¹ utilisées par ces outils, on note premièrement beaucoup d'outils orientés desktop développés avec le langage python. Ce langage est aussi utilisé dans le cadre de plateforme web avec une architecture client/serveur. Les Frameworks python de développement web Django et Flask sont alors principalement utilisés pour le développement côté serveur. Cela peut s'expliquer par la prédominance du langage python dans les Frameworks utilisés en data-science/machine learning.

On note aussi, évidemment, une part importante de projet en javascript pour le développement d'interface de saisie côté client web. A noter que **certaines applications offrent à la fois une interface web et desktop via l'utilisation du Framework electron.js** qui permet de développer des applications "de bureau" en javascript en encapsulant le moteur de rendu V8 web chrome dans l'application.

Enfin on peut souligner que certains outils offrent un modèle mixte open-source/propriétaire, avec la mise en open-source d'une partie de leurs outils et la commercialisation de service ou de version plus évoluée de l'outils (<https://labelstud.io/> ou bien [coco-annotator](https://coco-annotator.com/))

¹ ensemble des outils technologiques utilisés pour développer et faire fonctionner un programme

Les outils open-sources sont globalement moins complets que les plateformes propriétaires et souvent plus centrés sur la partie labellisation et moins sur les aspects de qualités, gestion de production et/ou saisie collaborative.

Besoins et existant IGN en labellisation de données

L'étude s'est principalement portée, via la recherche bibliographique et les entretiens sur les projets et travaux IGN qui ont déjà identifiés des besoins en labellisation de données d'apprentissage. Toutefois comme l'utilisation de l'IA est émergente à l'IGN ces besoins ne peuvent prétendre faire le tour des besoins IGN à moyen terme sur ce sujet. Il s'agit ici plutôt des besoins urgents ou à court terme (1 ou 2 ans). Une mise à jour de cette étude semble ainsi pertinente après ces échéances proches passées ainsi qu'un approfondissement des solutions algorithmiques et briques informatiques open-sources existantes.

Besoins projets IA

L'IGN est actuellement engagé dans plusieurs projets ayant une forte composante IA :

- Automatisation de la production de carte d'occupation du sol grande échelle (FTAP suivi de l'artificialisation avec la DGALN)
- Mise à jour et évolution de la BDForêt (étude interne IGN)
- Expérimentation sur les données Lidar aériens dans le cadre du FTAP Lidar-HD
- Expérimentations dans le cadre du monitoring PAC (extraction d'éléments de paysages et contrôle de l'activité agricole)
- Projet CarHab

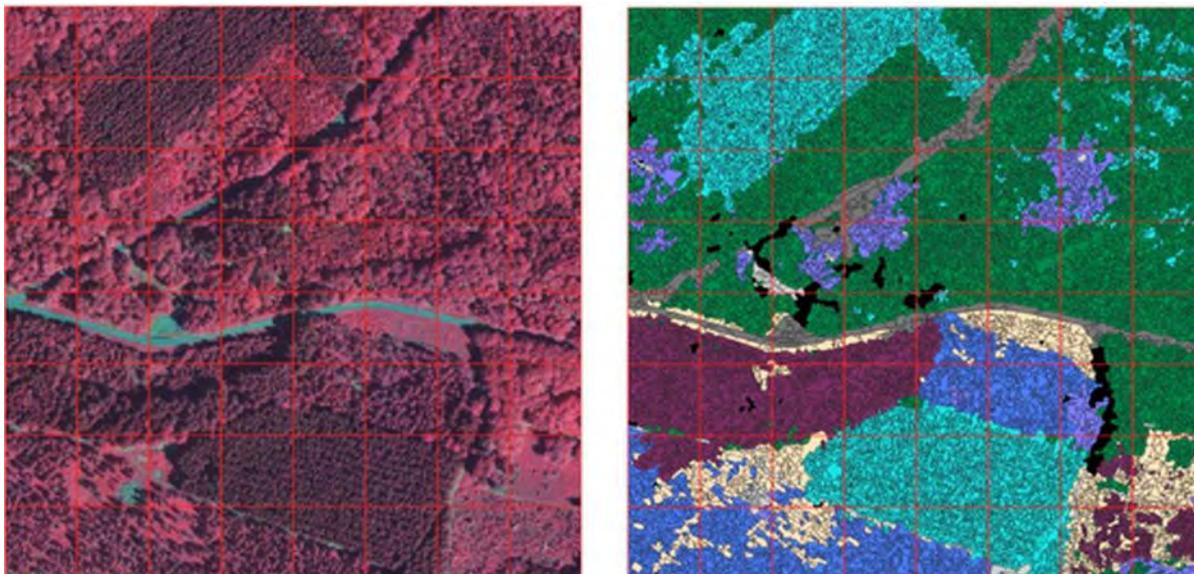
Chacun de ces projets nécessite des données d'apprentissage afin d'alimenter les processus IA associés. Toutefois dans le cadre des projets CarHab et des expérimentations sur le monitoring PAC ces données d'apprentissage proviennent des partenaires et ne sont donc pas associées aujourd'hui à un besoin IGN. Parmi les autres projets il existe actuellement trois méthodes d'obtention des données d'apprentissage :

- Dérivation automatique des labels : à partir des bases de données existantes. Par exemple extraction, filtrage et rastérisation des bâtiments BDUi (BDTopo) pour correspondre à un millésime BDOortho.
- Constitution semi-automatique : à partir de classification pixel ou objet (après segmentation) sur de petites zones de traitements (quelques km²). Eventuellement des corrections et des retouches manuelles sont faites après les étapes de classifications.
- Saisie manuelle / labellisation : à partir de photo-interprétation sur images et pour l'instant via des outils SIG.

La partie de saisie manuelle pour les thématiques OCSGE et forêt est relativement importante et peut s'expliquer par plusieurs points :

- L'actualité de la donnée existante est trop différente du millésime de l'orthophotographie à utiliser pour ces projets.

- La donnée est trop généralisée et ne correspond pas alors à ce que les méthodologies actuelles d'apprentissage profond arrivent à extraire. On cherche aujourd'hui des labels plus proches du pixel image que de l'objet topographique/cartographique.
- Certains thèmes nécessaires pour l'occupation du sol ne sont couverts par aucune base de données existante et sont difficiles à classifier par des méthodologies d'apprentissage machine classique (non deep learning).



Exemple de labellisation faite pour la mise à jour BDForêt. On observe que la précision d'annotation est plus grande que celle disponible dans la BDForêt elle-même (© IGN)



Exemple de couche thématiques extraites pour l'OCS GE en zone urbaine (rouge bâtiment BDTopo, violet bitume par extraction BD+ complétion manuelle, vert foncé et vert clair extraction semi-automatique de végétation haute et basse, jaune matériaux minéraux/sols nus) (© IGN)

Le constat actuel est que la **saisie manuelle ou semi-automatique de ces échantillons est très chronophage** et donc coûteuse et que **l'outillage SIG qui sert aux productions actuelles IGN n'est pas forcément adapté à la demande d'une saisie précise au niveau du pixel** et avec un aspect topographique moins important.

Le besoin est donc à court terme de disposer d'outils de labellisation plus efficaces pour les projets OCSGE et BDForêt. Ces deux projets ont un besoin similaire de labellisation, appelé segmentation sémantique dans la communauté IA. De plus il est souhaité que l'outil conserve au maximum certaines des fonctionnalités des outils SIG qui sont jugés nécessaires ou très utiles lors de la photo-interprétation d'image aérienne :

- Visualisation d'image assez large (volume de données et nécessité de visualiser le contexte pour la photo-interprétation)
- Capacité à afficher des informations selon différents modes et à passer d'un affichage à un autre facilement. Par exemple visualisation IRC pour la photo-interprétation de la végétation, possibilité d'afficher plusieurs millésimes pour la saisie de changement/évolution ou pour lever des ambiguïté d'analyse. Cela doit pouvoir se faire rapidement (peu de clics) et de façon fluide (rapidité et réactivité de l'affichage).
- Gestion/conservation des données de géoréférencement.

En outre la possibilité d'afficher des séries temporelles, comme celles des satellites Sentinel-2 serait un plus pour certaines thématiques.

Il n'y a pas d'autres contraintes fortes sur l'outil de labellisation, pas de contrainte de processus collaboratif ou de développement open-source. Par contre il est préférable que l'outil reste si possible cohérent avec les technologies et processus actuels de la production IGN.

Besoins recherche IGN - LASTIG

Au niveau de la recherche IGN les besoins qui nous ont été remontés concernent aussi principalement la labellisation de données images. Il y a toutefois des différences notables avec les besoins exprimés par la production.

Premièrement en plus des besoins de labellisation sur les données images et lidar aérienne et satellites la recherche IGN a aussi un **besoin de labellisation sur les cartes anciennes**, par exemple dans le projet ANR Soduco. Ainsi des chercheurs IGN ont participé à l'élaboration du benchmark (et challenge IA associé) [ICDAR 2021 Competition on Historical Map Segmentation](https://icdar21-mapseg.github.io/tasks/task1/).



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

Ensuite la **recherche IGN s'inscrit fortement dans le cadre de la recherche reproductible** et des principes FAIR (Findable, Accessible, Interoperable, Reusable) que cela soit pour les données ou bien les outils nécessaires aux recherches et à la production de données associées. C'est aussi une contrainte de plus en plus forte au niveau des montages de projets européens et ANR. Cela

conduit donc à préférer le développement et l'utilisation d'outils de labellisation open-source par rapport à l'utilisation de plateformes propriétaires.

De plus certains axes de recherche pourraient porter sur **les méthodologies d'active learning**, la recherche des interactions hommes-machines (outils "*human in the loop*") pour la production des données géographique comme l'occupation du sol, ou encore sur les outils collaboratifs (édition ou contrôle). Dans ces cas-là, l'accès à des logiciels qui puissent être adaptés et modifiés est donc préférable et favorise de nouveau l'utilisation des outils ou plateformes de labellisation open-source.

Au niveau de l'ENSG et de l'enseignement les outils open-sources peuvent être intéressants dans le contexte des projets informatiques en fournissant une base sur laquelle les élèves peuvent essayer d'ajouter des fonctionnalités. Il est à noter que des élèves ont déjà participé à une "carto-party" organisée dans le cadre du projet LandSense. Toutefois il n'y a pas de besoin en outils ou plateforme de labellisation remontés dans le cadre de l'école. En effet d'une part la durée des cours et TD impose souvent de partir de données déjà préparées pour un exercice spécifique et d'autre part la labellisation de données en elle-même représente un intérêt pédagogique faible. La mise à disposition de tels outils pourrait par contre servir à quelques élèves dans le cadre des stages de fin d'étude sur certaines problématiques orientés apprentissage machine. Enfin l'école via les travaux sur le terrain est plus intéressée par l'étude des outils de saisie et gestion de données terrain que par ceux de type photo-interprétation.

Enfin la variété des types de données utilisées par l'ENSG ou le LASTIG est plus grande que celle utilisée en production IGN aujourd'hui, notamment au niveau des données lidar ou bien des séries temporelles d'images satellites sentinel-2 et sentinel-1 ou encore sur les données aériennes de type multi ou hyper spectrale.

Outils IGN pour la saisie de données géographique.

Il est naturel de regarder si les outils de saisie de données géographiques utilisés et/ou développés par l'IGN peuvent répondre au besoin de labellisation des données d'apprentissage IA, comme il s'agit de logiciels déjà connus et maîtrisés par l'IGN.

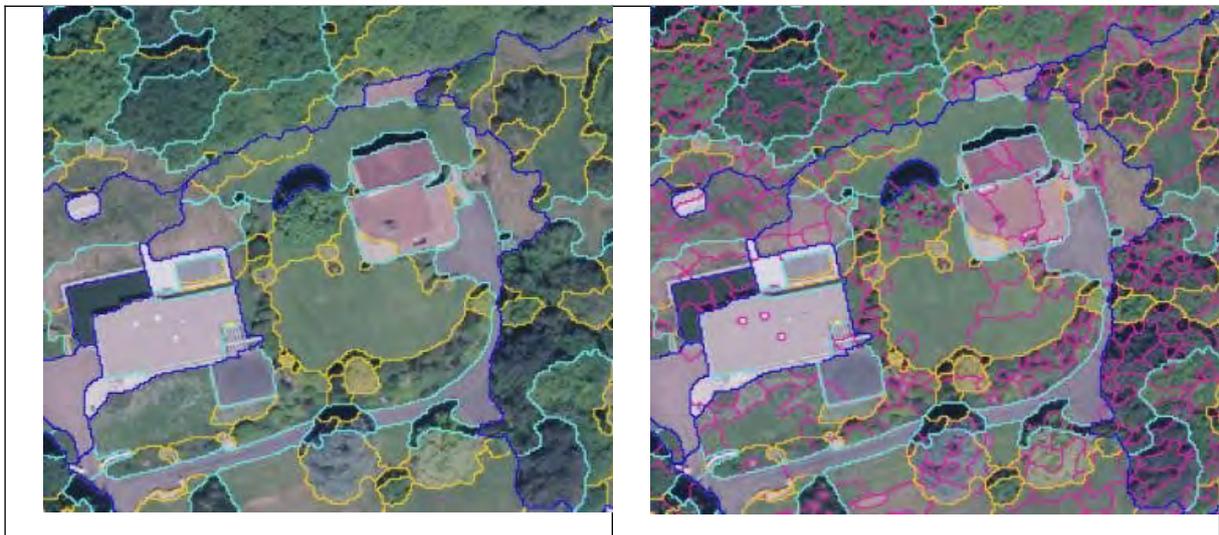
Logiciels SIG desktop

Au niveau de la production de données les logiciels utilisés sont principalement des outils SIG classiques comme Geoconcept, ArcGIS et de temps en temps Qgis ou OpenJump. Selon les chaînes de productions des plugins/addins ont été développés sur ces logiciels pour rendre leur utilisation plus efficiente. Toutefois, aujourd'hui, aucun de ces logiciels n'offre de solutions intéressantes pour la labellisation de donnée d'apprentissage souhaitée par, entre autres, les projets OCSGE et BDForêt. En effet la labellisation de données d'apprentissage est faite avec une précision très grande (proche du pixel) alors que les outils de saisie/découpe/modification de polygones présents dans les SIG conviennent mieux pour de la saisie de données plus généralisées par rapport à la vérité image, comme les données topographiques ou cartographiques.

Une solution, actuellement en cours de test est alors de développer des plugins spécifiques à ces SIG pour la saisie de données d'apprentissage sur image. Un stage pour le développement d'un plugin Qgis pour la labellisation des données

d'apprentissage sur les classes herbacés/sols nus/ligneux en zones de montagne a été effectué en 2020 et a donné des premiers résultats intéressants. Ce plugin propose de choisir et calculer une segmentation image sur une zone choisie puis offre une interface de saisie conçue pour la labellisation efficace des segments/surfaces² obtenus. Lors d'une étude de l'existant pendant ce stage **il n'a pas été trouvé de plugin SIG offrant des fonctionnalités similaires à celles souhaitées pour les besoins OCSGE BDForet.**

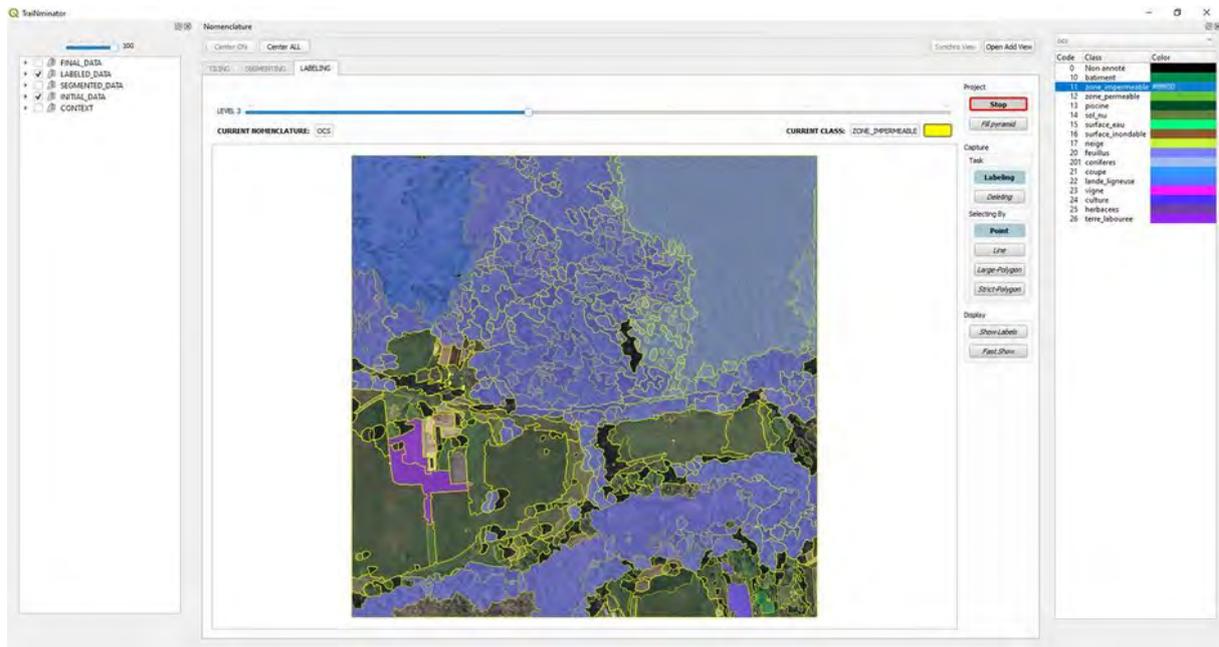
Les premières labellisations des données pour la mise à jour de la BDForet ont, elles, été faites avec les outils ESRI(©) ArcMap. Elles ont utilisé, pour accélérer la saisie, plusieurs couches résultantes d'une segmentation hiérarchiques des orthophotographie. Les différentes couches de segmentations permettant d'avoir un niveau de découpe grossier vers fins et s'emboîtant entre elles. L'affectation des segments à une classe de la nomenclature et le passage de label d'un niveau grossier vers un niveau plus fin étaient alors faits avec les outils existants par défaut dans ArcGIS. Le calcul de la segmentation a été faite avec le logiciel IGN Pyram qui implémente la méthodologie développée dans les travaux de [thèse de Laurent Guigue](#) (travaux de recherche IGN).



Exemple de segmentation hiérarchique obtenue par Pyram (© IGN) sur une image avec 3 et 4 niveaux affichés

Ces deux expérimentations ont conduit au développement d'un prototype de plugin Qgis plus adaptés pour la labellisation des données d'apprentissage OCSGE-NG et BDForet. Ce plugin est en cours de test et finalisation par le service des développements métiers (SDM). Il propose, comme pour les expérimentations BDForet, d'annoter des couches de segmentation hiérarchique mais offre par contre une interface graphique plus adaptée et efficace pour cette tâche que les outils disponibles dans Qgis.

² Le résultat d'une segmentation est ici constitué de polygones (donc de surfaces) mais ceux-ci sont souvent appelés objets ou segments pour indiquer qu'ils sont le résultat d'une segmentation et non d'une saisie manuelle. Il ne s'agit donc pas de segment au sens ligne entre deux points/sommets.

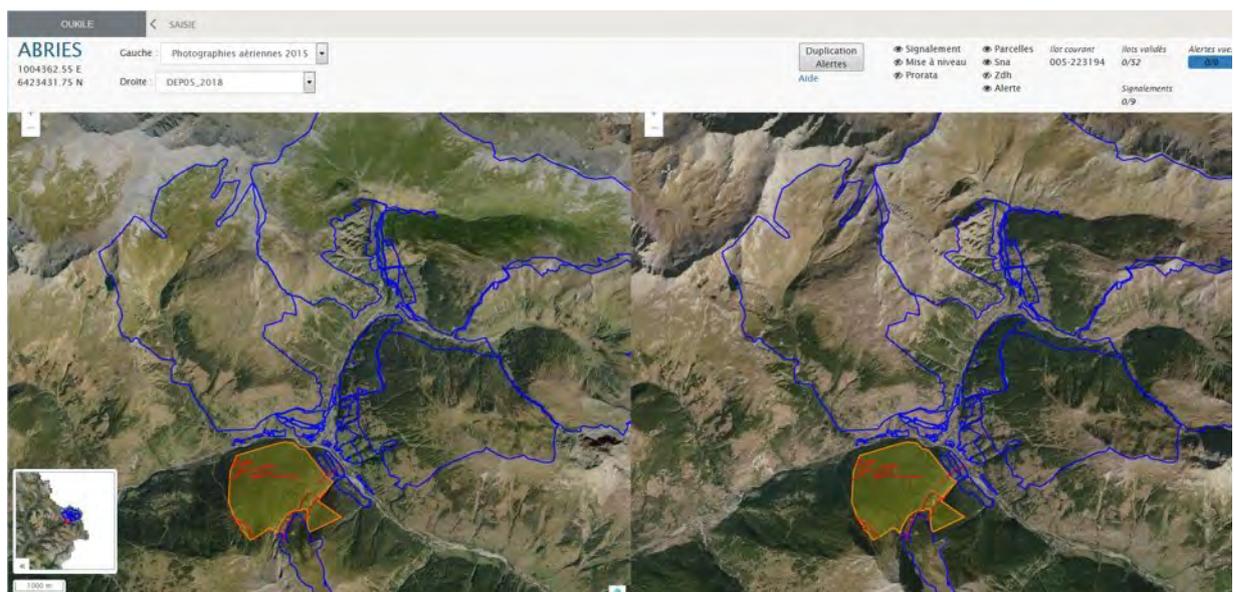


Visualisation du plugin QGIS de labellisation de données développé par le SDM (© IGN)

Outils web collaboratifs

L'IGN a également développé et utilisé des outils de saisie de données avec des outils web et des fonctionnalités collaboratives.

Il y a d'une part les outils *Oukilé* et *Spot-it* développés pour remonter des alertes de détection de changement par photo-interprétation. Dans le premier cas, *Oukilé*, il s'agit de créer les alertes de modifications de données entre deux millésimes ortho pour la production des mises à jour des données RPG. Cet outil reprend du code et des fonctionnalités de *Spot-it*, application qui a été développée pour remonter des détections de changement entre deux millésimes ortho ou satellites sur des thèmes choisis (végétation, bâtiment, route etc..). Dans le cas de *Spot-it* l'annotation de changement était faite à l'image (tuile de taille configurable) alors que pour *Oukilé* il existe en plus des fonctionnalités de saisie de type signalement.



interface de l'outil Oukilé (© IGN)

Ces outils proposent une interface de saisie pour des alertes mais sont plutôt spécialisés sur des cas spécifiques. De plus la configuration des différents chantiers (base de données) est faite par des scripts qu'un administrateur doit lancer selon une procédure définie et non pas une interface ou API accessible côté client telle que les plateformes de labellisation IA le proposent.

Une autre famille d'outils disponibles à l'IGN sont ceux associés à l'espace collaboratif (<https://espacecollaboratif.ign.fr/>). Leur objectif est d'une part d'offrir un moyen de remontés des signalements (erreurs, mises à jour) sur des bases de données géographiques, en particulier sur celle de l'IGN comme la BDTopo. Cela est fait via le développement d'un service offrant une API de signalement et plusieurs clients (web ou plugin SIG) utilisant cette API. D'autre part ces outils permettent aussi d'offrir des fonctionnalités d'éditions collaboratives directes sur les données géographiques, cela est fait principalement par le développement d'applications (guichets) spécifiques à un besoin particulier.

Afin de mettre en place une application de type guichet collaborative il faut actuellement :

- Installer et configurer préalablement la base de données géographiques sur laquelle on souhaite faire de l'édition collaborative
- Installer et configurer des flux WMS s'ils ne sont pas déjà disponibles (via les géoservices IGN par exemple)
- Développer une interface utilisateur cliente dédiée au guichet.

Par comparaison aux plateformes de labellisation IA les outils collaboratifs IGN n'offrent donc pas d'outils d'édition spécifique mais ceux-ci peuvent être développés avec l'application cliente. De plus il n'y a pas non plus de fonctionnalités de gestion de production et de contrôle qualité intégré.

Certaines applications d'éditions collaboratives ont été développées et utilisées dans le cadre du projet LandSense pour l'édition de données de types OCSGE mais peuvent être difficilement reprises. D'une part les outils des partenaires comme Geo-Wiki ne sont pas open-source et donc l'IGN ne peut déployer lui-même une plateforme similaire. Et d'autre part les outils développés par l'IGN, comme l'application paysage, ont utilisé une version spécifique des codes de l'espace collaboratif adaptés pour ce projet.

Enfin les codes de l'espace collaboratif ne sont pas open-source actuellement et sont assez dépendant des choix du SI interne IGN et donc ne sont pas conçus pour être génériques et pouvoir être déployés facilement hors de l'IGN.

Quelques pistes de réponses aux besoins

L'étude de l'existant et des besoins IGN ne fait pas apparaître de solution unique et immédiate aux différents besoins de l'IGN. Par exemple les outils collaboratifs IGN ne semblent ni pouvoir répondre rapidement au besoin du LASTIG ni être compatibles avec certains critères comme la distribution du code en open-source et la reproductibilité du déploiement.

On peut toutefois proposer quelques pistes d'actions qui pourraient aider à répondre à un ou plusieurs des besoins IGN.

- Développement de prototypes desktop d'interface de labellisation. Le but est alors de tester rapidement certaines méthodologies de saisie, comme celle basée sur une segmentation hiérarchique et de disposer de premiers outils pouvant "dépanner" pour certains besoins de production ou recherche IGN. Au vu de l'existant SIG et des outils de de labellisation open-source, des développements en python semblent être les plus appropriés, soit sous forme de plugins SIG ou bien d'application indépendantes. De plus pour la recherche il faudrait diffuser ces outils en open-source.
- Développement de méthodologie de labellisation choisies pour les clients web. L'existant au niveau des outils et plateforme de labellisation de données d'apprentissage ainsi que l'évolution des outils IGN montrent que le développement de solutions web est nécessaire à la création d'outils performants pour la labellisation IA. Comme ces développements web peuvent être plus coûteux et complexes que le développement de prototypes python il s'agit ici de porter les méthodologies testées et éprouvées par les POC python précédents dans des bibliothèques pouvant être réutilisées dans les différents développement de clients web à l'IGN. Cela pourrait être fait par la création de bibliothèques de type plugin ajoutant ces fonctionnalités de saisies de données d'apprentissage aux bibliothèques javascripts de type Leaflet et Openlayer (utilisées dans les développements web sig).
- Mise en place de prototype web de labellisation pour les besoins recherche et innovation. Cela permettrait de répondre rapidement à certains besoins de la recherche et de gagner en compétence et expérience sur la question. Les expériences concluantes pourraient alors être proposées comme évolution pour les outils internes IGN. Quelques thématiques de labellisation qui semblent intéressantes à pousser aujourd'hui et devraient être réalisables à court/moyen terme sont :
 - o La saisie de polygones /vérité terrain pour les classifications d'occupation du sol à partir de données Sentinel. La classification étant faite aujourd'hui par des algorithmes non deep learning cela demande moins de saisie de données et apporte un intérêt plus rapidement.
 - o La vectorisation de carte ancienne, via l'utilisation et l'adaptation des outils OSM. Cela permettrait d'avoir un retour plus poussé sur les avantages et limites de ces outils et leur adaptabilité à un besoin spécifique.
 - o Le contrôle qualité de résultats de classification.
- Test de plateformes open-sources ou propriétaires. Le test de plateformes propriétaires est intéressant pour faire des retours d'expérience à la production IGN alors que l'essai et le déploiement de solutions open-sources permettrait de mieux juger leur maturité (difficulté de prise en main et installation) et du reste à faire pour les adapter au besoin des données géographiques. Un exemple de plateforme open-source qui semble intéressant à tester car similaire à nos besoins pourrait être <https://github.com/biigle> qui est utilisée pour l'annotation d'image/faune marine ou bien l'outil labelstudio (© Heartex) qui fonctionne avec une offre communautaire open-source et une offre propriétaires plus complète.

2 État de l'art des outils de labellisation pour données géomatiques

Le succès rapide et grandissant des technologies deep learning a conduit à un besoin important en termes de labellisation de données afin d'alimenter celles-ci en données d'apprentissage. En effet, les architectures de réseaux de neurones profond nécessitent historiquement de larges volumes de données pour pouvoir être entraînées. On peut noter que cela s'est quelque peu atténué aujourd'hui avec les méthodologies d'apprentissage semi/faiblement/non supervisées mais il reste quand même un besoin conséquent en labellisation de données.

L'obtention et la constitution de ces différentes données d'apprentissage étant relativement coûteuses, de par le volume de données associé et le détail de saisie parfois nécessaire, cela a conduit à la création de multiples plateformes et outils de labellisation, dont le but est d'offrir des outils ou services permettant de diminuer le coût d'obtention de données d'apprentissage. D'une part, un certain nombre d'outils, essentiellement open-source, ont été créés par et pour la communauté des chercheurs utilisant les algorithmes de deep learning afin d'aider à la constitution de jeux tests. Et d'autres part, des plateformes et outils propriétaires sont apparus afin de proposer des services de labellisation aux différentes entreprises souhaitant utiliser des méthodologies deep learning dans leur propre processus. Le choix entre re-développer son propre outil de labellisation, utiliser ou adapter un outil open-source existant ou bien utiliser une solution sur étagère est alors très dépendant des avantages/inconvénients de chacune de ces stratégies et du besoin spécifique en données d'apprentissage.

Une part essentielle de ces outils est l'interface utilisateur de saisie des labels proprement dite. En effet, au vu du volume de données à saisir, l'efficacité de cette dernière, par exemple en nombre de clics nécessaires pour saisir un label ou en fluidité d'affichage, est fortement liée au coût final de la production de données d'apprentissage. Les autres fonctions rentrant en jeu dans ce coût sont plutôt les fonctionnalités de gestion de production et suivi/contrôle de la qualité. Enfin, la plupart des outils et plateformes existants ont été créés pour des besoins assez classiques comme le traitement de texte (analyse, traduction) ou le traitement d'images "terrestres/naturelles" (photos Instagram, Facebook, véhicule autonome) et ne sont pas toujours adaptés à certains cas de données très spécifiques comme cela est le cas dans les domaines des images de télédétection ou bien des images médicales.

La suite de cette partie présente un état de l'art des différents outils de labellisation existants, il n'est toutefois pas générique et se concentre essentiellement sur le cas de la labellisation de la donnée image qui représente la plupart des données de labellisation en télédétection/géomatique. De plus nous détaillons à la fois des solutions commerciales et open-sources car selon les projets, qui sont soit plus orientées production/industrialisation ou bien soit plus recherche et développement, chacune a des avantages/inconvénients respectifs. Enfin, nous présentons de façon détaillée les fonctionnalités de saisie existantes et leurs intérêts selon les cas d'utilisation deep learning. Cela peut permettre de comparer différentes solutions selon un problème spécifique ou bien de guider la définition des besoins dans le cadre d'un développement d'outils en interne.

Fonctionnalités générales des outils et plateforme de labellisation

Dans un premier temps nous présentons les principaux éléments clefs permettant de catégoriser les différents outils et plateformes de labellisation. Nous détaillons les grandes familles de fonctionnalités offertes par ces outils et listons les principales sous-fonctionnalités associées trouvées lors de notre recherche d'état de l'art ainsi que les éventuelles particularités ou exemples associés aux domaines de notre étude (télé-détection, cartographie, ...).

Principales caractéristiques et fonctionnalités

Les principaux éléments clefs des outils et plateforme d'annotation peuvent être résumé par :

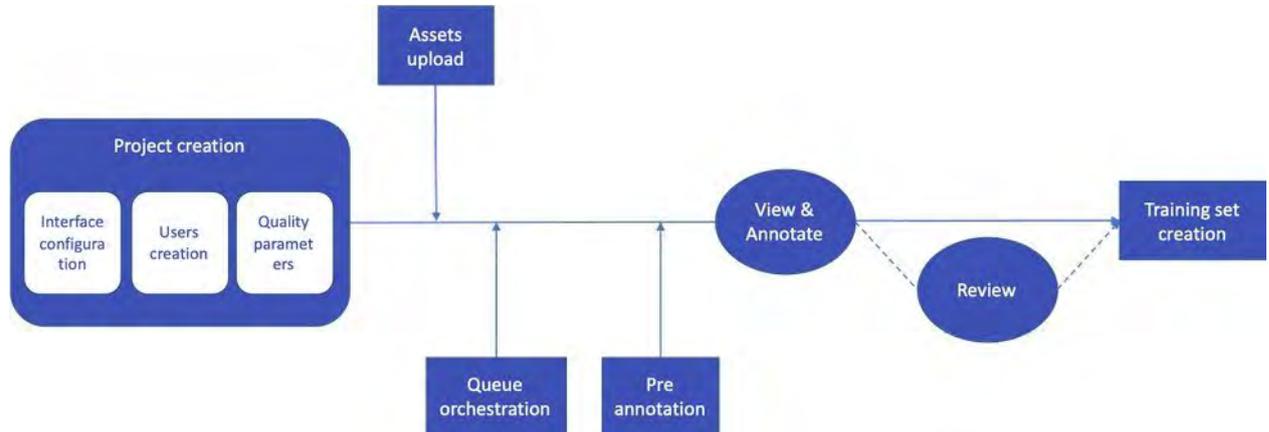
- Méthodes d'annotation
- Types de données supportés
- Licence et adaptabilité
- Mode de déploiement

Les fonctionnalités supplémentaires que peuvent avoir les outils et plateformes sont :

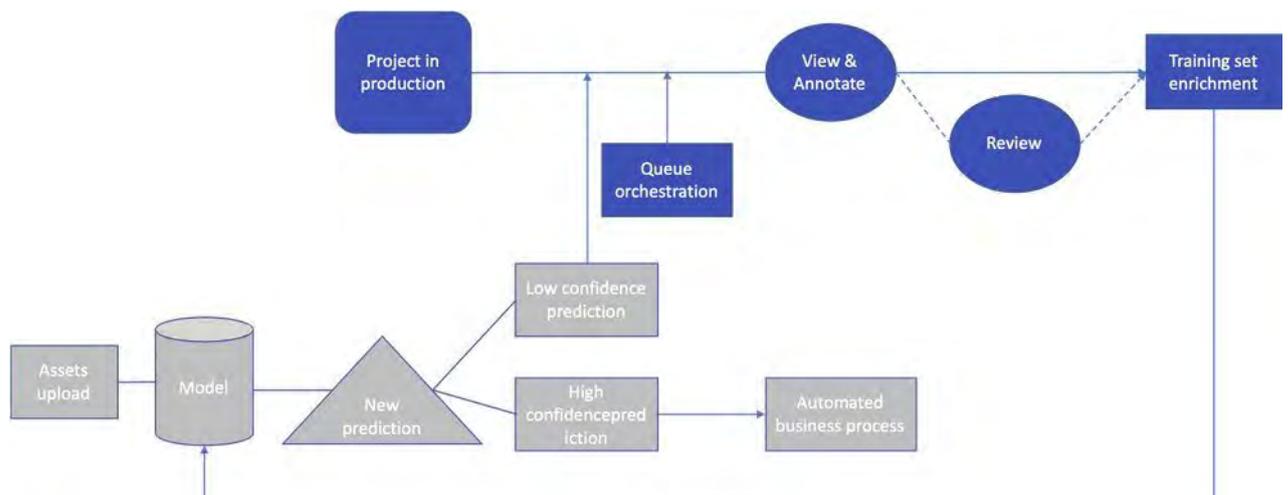
- **Gestion des jeux de données** : cela comprend le stockage, l'accès et la diffusion ainsi qu'éventuellement des notions de versionnement des jeux de données créés.
- **Configuration de projet** : spécification d'une nomenclature, des rôles et accès des différents utilisateurs, import/liens vers un jeu de données à labelliser ...
- **Gestion de la qualité** : ensemble de fonctionnalités permettant de garantir la qualité de la labellisation. Cela est principalement fait par l'intégration de processus d'annotation avec une étape de validation des données saisies.
- **Suivi et gestion de production** : affichage sous forme de tableaux de bord de l'avancement des différentes tâches sur un projet de labellisation. Les fonctionnalités peuvent être plus ou moins avancées et permettre de visualiser des temps de labellisation par opérateur ou type de données.
- **Sécurité** : garantie de la propriété et non diffusion/fuite des données brutes et labellisées utilisées dans la plateforme. Cela comprend la sécurité au niveau des protocoles d'échanges et la compatibilité de l'hébergement des données avec des contraintes de type RGPD.
- **API /SDK** : possibilité d'automatiser et d'intégrer finement les outils de la plateforme au reste des processus via des appels à une API ou à un SDK de développement.
- **Automatisation** : outils permettant d'automatiser une partie des tâches de labellisation. Cela est lié à la gestion de production quand il s'agit d'orchestrer des chaînes de traitements. Plusieurs degrés d'automatisation existent :
 - o Niveau 0 : annotation manuelle pour toutes les tâches
 - o Niveau 1 : annotation assistée pour diminuer le nombre de clics.
 - o 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.

- o Niveau 3 : pré-annotation quasi-complète. L'opérateur fait essentiellement du contrôle qualité de l'annotation automatique et saisie de cas complexes.

On peut voir certaines des étapes associées à ces grandes fonctionnalités dans les schémas de chaîne de labellisation proposée par la société kili-technology lors des étapes de création et déploiement d'un modèle IA.

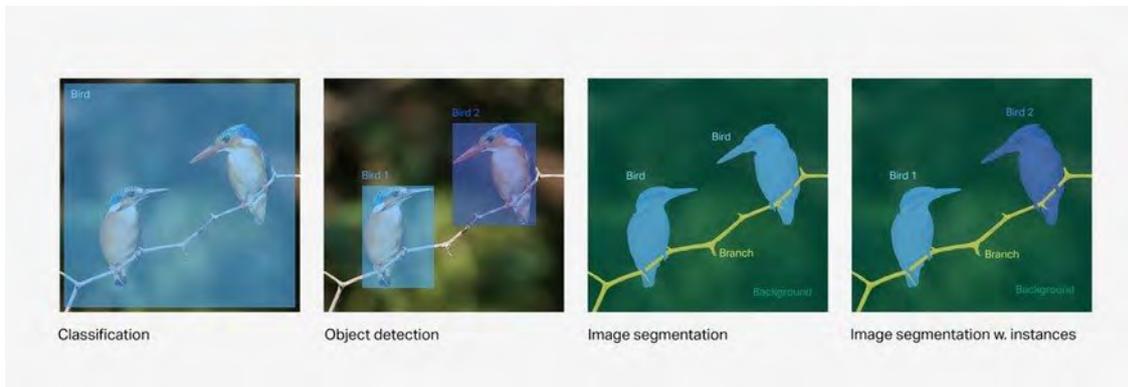


Fonctionnalités utilisées lors de la phase d'entraînement/création de modèle IA (© kili-technology)



Fonctionnalités utilisées lors de la phase de déploiement et entretien de modèle IA (© kili-technology)

Au niveau des données géomatiques **les tâches principales sont celles de segmentation (sémantique, instance ou panoptique)**. Les outils et plateformes de labellisation ne supportant pas la donnée image et n'offrant pas d'outils pour faire de la labellisation sémantique sont à exclure de la liste des outils intéressants dans les domaines géomatiques et télédétection. Cela écarte déjà un certain nombre d'outils qui se spécialisent dans les tâches de classification ou détection d'objets plus courantes dans les domaines de l'analyse d'images "terrestres".



Type de labellisation image (© Labelbox : <https://labelbox.com/blog/august-highlights-from-the-labelbox-team/>)

Gestion des jeux de données

Les outils de labellisation ayant besoin d'avoir accès aux données que les utilisateurs souhaitent labelliser, une fonctionnalité importante est celle de la gestion des formats d'entrée et sortie supportés par la plateforme ainsi que le type de stockage auquel elle permet d'accéder.

Les type de stockages qui peuvent être proposés sont :

- **Dossiers/liste de fichiers sur un stockage disque**, principalement pour les outils à installer sur son ordinateur propre.
- **Import/transfert des données depuis un stockage local** vers le stockage cloud associé à la plateforme et au projet.
- **Import/transfert des données depuis un stockage cloud**. Souvent depuis un stockage supportant le protocole S3 d'Amazon (stockage objet).
- **Import à partir d'url (visualisation)**. Dans ce cas l'outil n'héberge pas les données mais se sert d'urls pour accéder à un service de visualisation des images. Cette fonctionnalité peut garantir une meilleure sécurité des données mais par contre empêcher d'utiliser des fonctionnalités avancées d'automatisation ayant besoin d'avoir accès localement à la donnée pour faire des calculs.

Au niveau des plateformes de labellisation la plupart de ces fonctionnalités sont disponibles via une API ou un SDK afin de pouvoir s'adapter plus facilement aux différents cas d'usage utilisateur. Ce dernier doit alors écrire du code permettant de faire le lien entre ses données (organisation, format) et ce qui est attendu ou livré par la plateforme.

Les formats de données images supportés sont assez restreints et sont principalement ceux des images RVB (PNG, JPG et parfois TIFF), de plus la taille des images est souvent limitée à quelques mega-pixels. Cela peut donc poser problème pour la gestion des images multi-spectrales et temporelles ainsi que pour les images aériennes ou spatiales qui sont souvent plus grande que la taille limite de ces plateformes. Il faut alors soit pre-tuiler les données images à labelliser ou alors utiliser un des outils supportant les flux WMS ou autre protocole de visualisation de données images. Chaque outil et plateforme possède généralement son propre format de stockage des données labellisées ([par exemple basé sur json pour superannotate](#)) et propose à partir de ce format

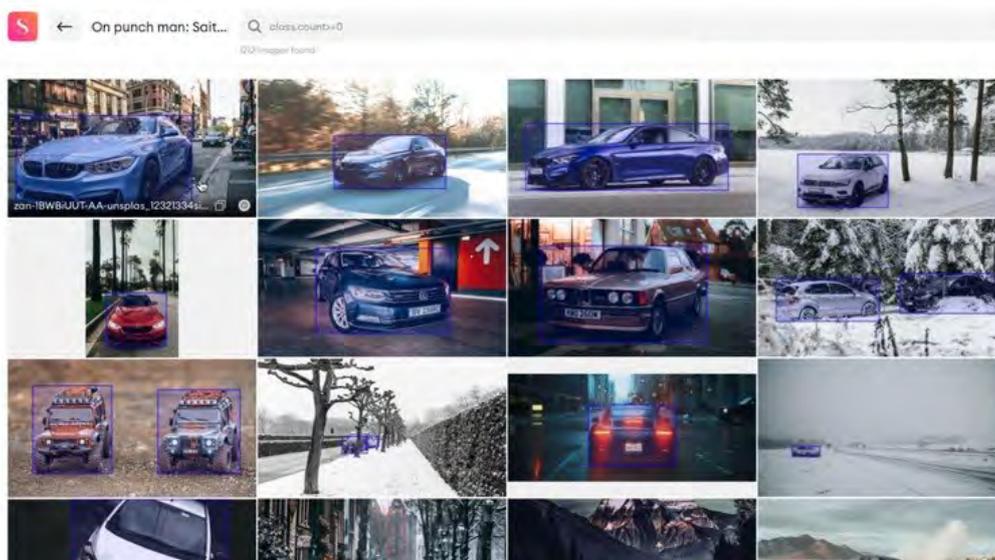
des conversions vers les formats standards d'annotation en apprentissage profond comme COCO, VOC ou YOLO.

Les données à annoter et les jeux de données sont la plupart du temps stockés sur forme de base de données et chaque objet possède plusieurs attributs qui peuvent par exemple refléter son statut ou bien les labels auxquels il est associé, le nombre d'annotateurs, le temps de saisie, etc...

Quelques exemples de statut pouvant être associés à une donnée sont :

- "à labelliser"/"not started"
- "en labellisation"/"in progress"
- "en contrôle"/qa
- "Approuvé"/completed
- "Éliminé"/skipped
- "Rejeté"/returned

Ces différents attributs sont alors utilisés pour fournir des fonctionnalités de filtrage et sélection dans les jeux de données. Cela permet par exemple de visualiser rapidement des exemples associés à un label ou bien ceux ayant demandé le plus de temps de saisie. Les images sélectionnées peuvent être ensuite affectées à certaines tâches (contrôle, labellisation par un autre annotateur, ...)



exemple d'interface de visualisation/exploration des labels (© superannotate)

Enfin, dans les plateformes les plus complètes la gestion des jeux des données comprend aussi le versionnement des annotations et le partage d'image/labellisation entre différents jeux de données et entraînement de modèle deep-learning/projet.

Configuration projet et adaptation de l'interface

Les fonctionnalités de configuration et personnalisation des outils et plateformes sont assez diverses et vont des fonctionnalités minimalistes à des configurations plutôt poussées.

Gestion des rôles/permissions

Un des premiers points de la configuration d'un outil de labellisation est l'affectation des droits utilisateurs. Cela passe par l'affectation de différents rôles prédéfinis aux personnes et à une gestion des permissions par rôle et projet. Certains outils et plateformes ne proposent qu'un ou trois rôles tandis que d'autres proposent jusqu'à au moins sept rôles différents :

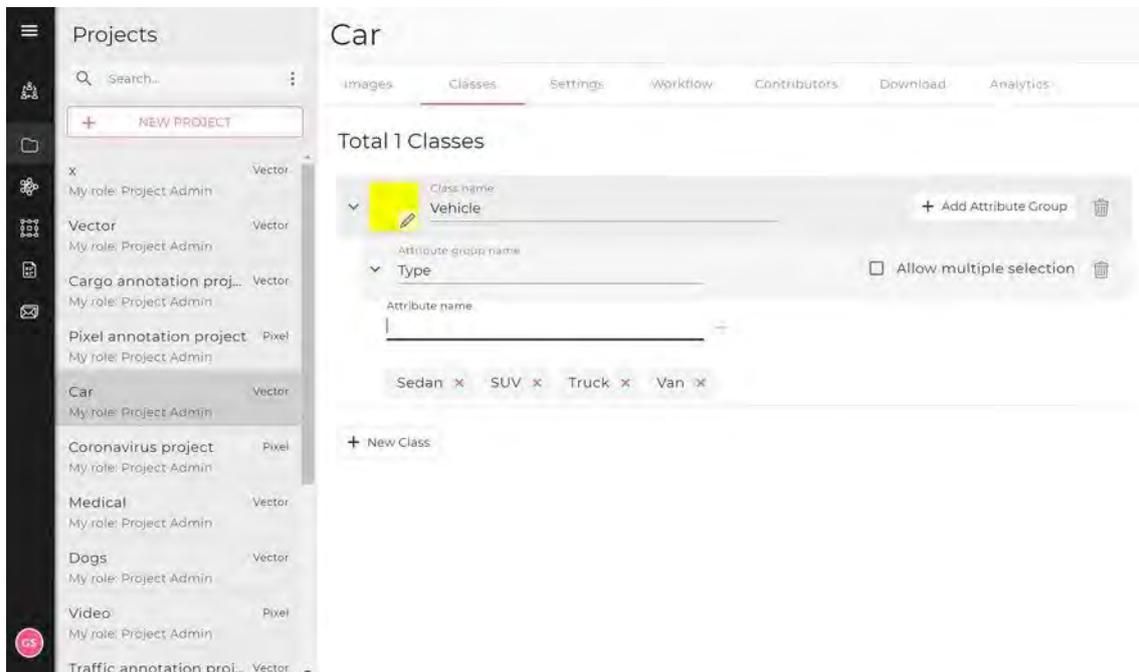
- **(Team) owner / admin** : propriétaire du compte sur l'outil ou la plateforme d'annotation. Possède l'ensemble des droits, il peut inviter des personnes à participer au projet et leur affecter l'ensemble des autres rôles. Il s'agit de la personne qui gère les droits ayant une incidence sur le coût facturé (par exemple l'accès au "token" pour accéder à l'API et sur les droits d'entraîner des modèles et d'accéder à des ressources de calculs...).
- **Team admin / developer** : administrateur/gestionnaire de compte sur la plateforme. Invite des personnes, leur affecte des rôles et les assigne à un ou plusieurs projets.
- **Project admin / manager** : responsable d'un projet de labellisation. Gestionnaire de production. Il a accès à l'ensemble des données du projet ainsi qu'aux outils de contrôles qualité et aux tableaux de bord (dashboard). Il est responsable de la configuration du projet (création des classes/nomenclatures entre autres).
- **Annotator** : personne en charge de labelliser les données. Ne peut voir et labelliser que les données associées aux tâches qui lui sont affectées.
- **Qa / reviewer** : personnes responsables du contrôles qualité. Peuvent voir l'ensemble des données des projets sur lesquels elles sont responsables qualité.
- **Customer** : client ayant commandé la labellisation des données. Distinction utile lors de sous-traitance de la labellisation. Peut voir le résultat de la labellisation.
- **Viewer** : Personne pouvant voir et éventuellement télécharger/accéder aux données labellisée. Utile dans le cadre d'un jeu de données "open-data" ainsi que pour toute communication interne et besoin d'illustration.

Configuration de projet

Le deuxième niveau de configuration est celui du paramétrage d'un projet de labellisation de données. Cela peut comprendre selon les outils :

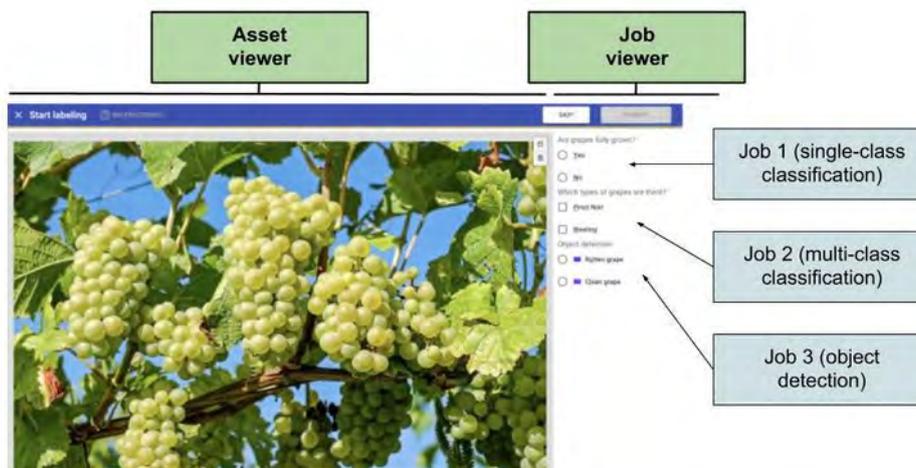
- **Choix de la/des tâches de labellisation** : par exemple choix entre classification ou segmentation sémantique. Les tâches peuvent être combinées et on peut demander de faire une détection d'objet ou une labellisation sémantique en l'associant à une tâche de classification image permettant par exemple de tagger une image comme "floue"/"nuageuse"/etc..
- **Création / affectation à un jeu de données** : spécifie un jeu de données à labelliser. La plupart du temps un élément du jeu de données constitue une tâche de labellisation.
- **Affectation à une équipe (teams)** : la gestion des rôles est définie ici ou bien héritée de celle de la définition d'une équipe configurée au préalable.

- **Création / affectation d'une nomenclature / ontologie** : création d'une liste de classe. Choix de label, couleur, éventuellement définition. Selon les outils la nomenclature peut être plus ou moins élaborée avec par exemple une notion de hiérarchie de classe ou bien associé à des types/attribut.



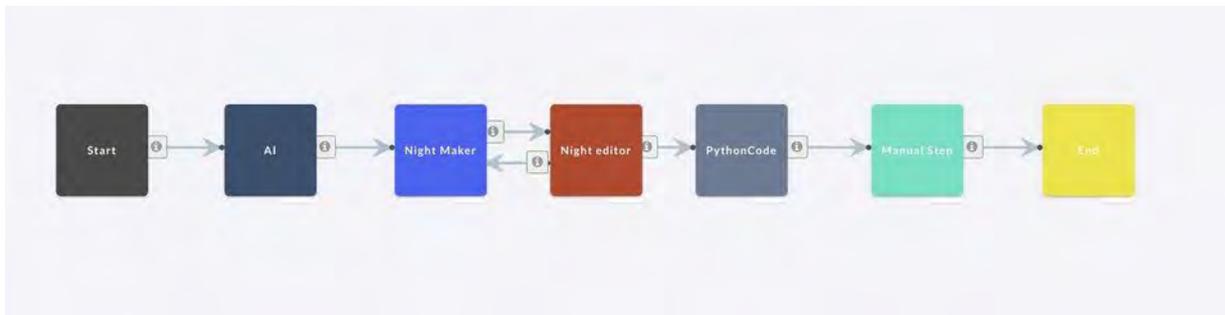
Étape de configuration des classes/nomenclature dans superannotate (©)

- **Configuration / personnalisation de l'interface de labellisation** : la configuration de l'interface de labellisation est soit faite au moment de choisir la tâche de labellisation ou bien peut être configurée de façon plus avancée en spécifiant un type d'interface selon les tâches à réaliser et en les positionnant les unes par rapport aux autres.



Exemple de personnalisation/configuration de l'interface de saisie ([kili-technology](http://kili-technology.com) ©)

- **Gestion qualité et workflow** : Certains outils permettent de choisir un processus qualité à associer au projet. Dans certains cas cela est fait via une configuration plus générique du workflow des jobs/tâches en configurant pour chaque donnée son cycle de vie.



Etape de workflow possible dans l'outils de playment.io (©)

- **Association à un modèle de prédiction :** dans le cas des outils ayant des fonctionnalités d'automatisation de la labellisation un projet est associé un à modèle de prédiction des labels.

Les modèles de données associés aux outils de labellisation peuvent être plus ou moins flexibles et permettre ou empêcher de réutiliser des données entre projet de labellisation. Par exemple la gestion de la définition des classes (aussi appelée nomenclature/ontologie) est souvent à faire pour chaque projet de façon indépendante, mais certaines plateformes rattachent plutôt cela au propriétaire ou à l'équipe de labellisation et permettent alors de **réutiliser une nomenclature sur plusieurs projets de labellisation**. La même chose se retrouve au niveau de la gestion des données, elles sont parfois affectées "physiquement" à un projet mais dans d'autres cas les données à labelliser peuvent être partager entre projets et jeux de données.

Il faut noter que la plupart du temps un élément du jeu de données à labelliser est considéré comme un job/une tâche de labellisation. Il n'y a pas toujours dissociation de ces deux notions même si cela pourrait être utile sur les jeux de données avec de grandes images comme les données aériennes ou spatiales ou même médicales.

Gestion de la production et contrôle de la qualité

Un aspect mis en avant par la majorité des plateformes commerciales de labellisation de données est l'ensemble de leurs fonctionnalités permettant de garantir la meilleure qualité de labellisation possible. Celle-ci est garantie en grande partie par des processus de contrôle qualité mais certains outils proposent aussi des fonctionnalités en amont de la labellisation comme la création de tests ou d'instructions d'annotation ainsi que des fonctionnalités de consolidation de base de connaissance.

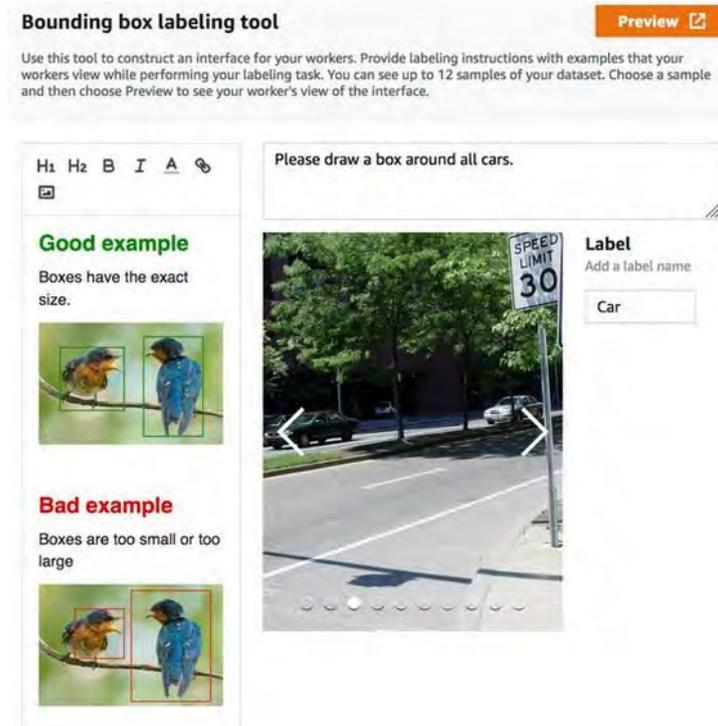
Tutoriels / guides et bases de connaissance :

Une première étape de contrôle de la qualité passe par une description claire des tâches à accomplir ainsi qu'une garantie que les annotateurs ont compris et maîtrisent ces instructions de labellisation. Dans cette catégorie de fonctionnalités il existe :

- **Création de test d'entrée pour les annotateurs :** chaque annotateur débutant doit alors commencer par la labellisation d'un ensemble de données test. Le résultat de labellisation est alors comparé automatiquement à une annotation faite préalablement par un expert ou la personne responsable des spécifications. Un exemple de mise en place de ce type d'outil est disponible dans la plateforme supervise.ly (Labelling

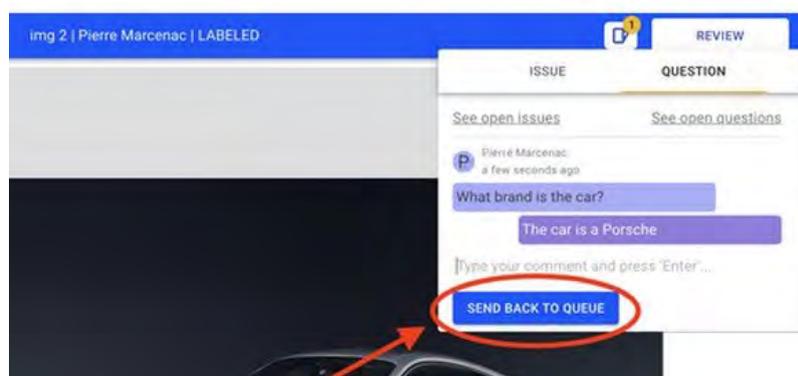
and report exam). L'examen peut aussi être fait par contrôle du responsable qualité qui valide les premières tâches de l'annotateur débutant.

- **Instruction de labellisation** : associe un document de spécification et/ou des exemples à une tâche de labellisation. Par exemple en associant un fichier PDF de guide d'annotation à un projet.



Instruction de saisie sur [amazon ground truth](#). "Good/bad example" (Amazon ©)

- **Questions / commentaire et base de connaissance** : fonctionnalité qui permet à un annotateur de poser des questions, principalement sur une précision de spécification, et au responsable du projet ou un responsable qualité d'y répondre. Ces questions/réponses peuvent être regroupées dans une base de connaissance commune au projet pour profiter à l'ensemble des annotateurs.

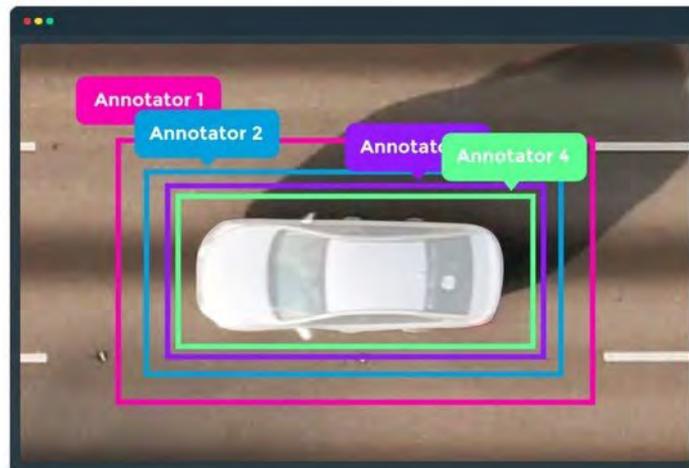


Fonctionnalités de questions/réponse [kili-technology](#) (©)

Outil de contrôle qualité :

Le contrôle qualité est fait essentiellement par plusieurs méthodes :

- **Benchmark / honeypot /gold standard** : annotations que l'on considère comme des vérités terrain et obtenues par annotation très précise ou après discussion/consensus entre plusieurs personnes expertes sur le sujet.
- **Consensus** : comparaison de plusieurs annotations sur la même donnée. Il peut être utilisé pour contrôler les annotateurs ou bien pour améliorer la qualité de labellisation sur les données difficiles à labelliser/interpréter et également obtenir une estimation de la qualité de saisie associée à chacune des classes de la nomenclature. Le consensus peut être activé sur un nombre fixe de données ou bien sur un pourcentage du jeu de donnée.



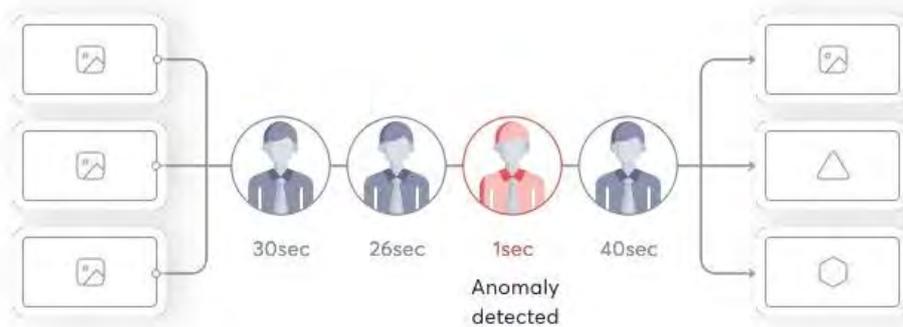
exemple de saisie multiple pour consensus (dataloop AI ©)

- **workflow de contrôle qualité** : une étape de contrôle est faite par une autre personne que l'annotateur pour chacune des données labellisées avant qu'elle soit validée.



[SuperAnnotate](#) : exemple de processus qualité pour la labellisation

Certaines plateformes proposent aussi des métriques liées aux annotateurs afin de repérer les anomalies parmi ceux-ci.



Détection d'anomalie (© super.ai)

L'ensemble de ces mesures qualité peut généralement être suivi via les tableaux de bord de suivi de production et d'avancement des tâches.

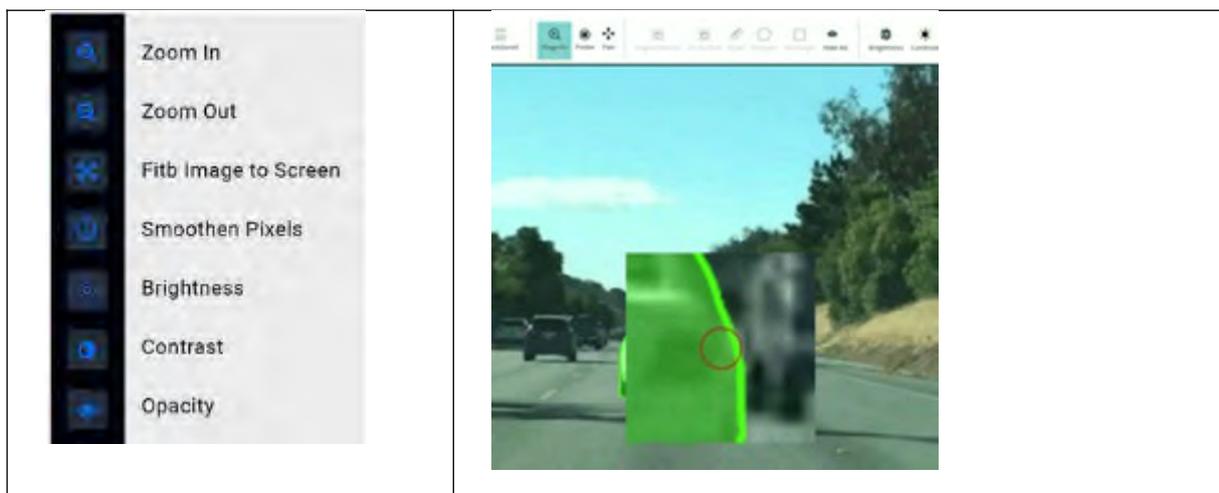
Fonctionnalités de saisies / interface utilisateur

La saisie de données étant le cœur des outils de labellisation, ces derniers disposent de fonctionnalités étendues dans ce domaine. De plus il s'agit de fonctionnalités ou l'interface graphique et l'efficacité de saisie est très importante. Lorsque l'on doit saisir 15000 données d'apprentissage un gain de 10 secondes par saisie de données représente déjà 42h de travail en moins, soit un gain d'une bonne semaine de charge par projet d'annotation.

Nous présentons ici les fonctionnalités concernant la labellisation de données images, et un peu lidar/3d mais ne détaillons pas les autres types de saisies existantes (audio, texte, vidéo).

Visualisation des données

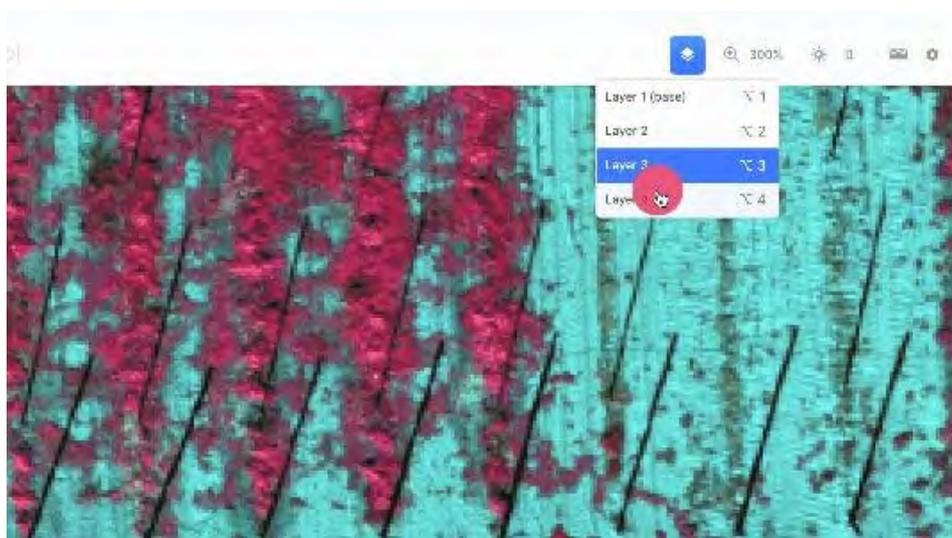
Les fonctionnalités de visualisation et navigation dans les images sont généralement assez basiques et comprennent les fonctionnalités de zoom/déplacement dans l'image, ainsi que quelques options de spécification de l'affichage (contraste, transparence de la couche d'annotation). Certains outils offrent des fonctions de visualisation plus évoluées comme une option loupe mais cela reste assez rare.



Outils de visualisation (gauche GT Tools © playment.io), droite outils [magnify](https://magnify.trainingdata.io) (© trainingdata.io)

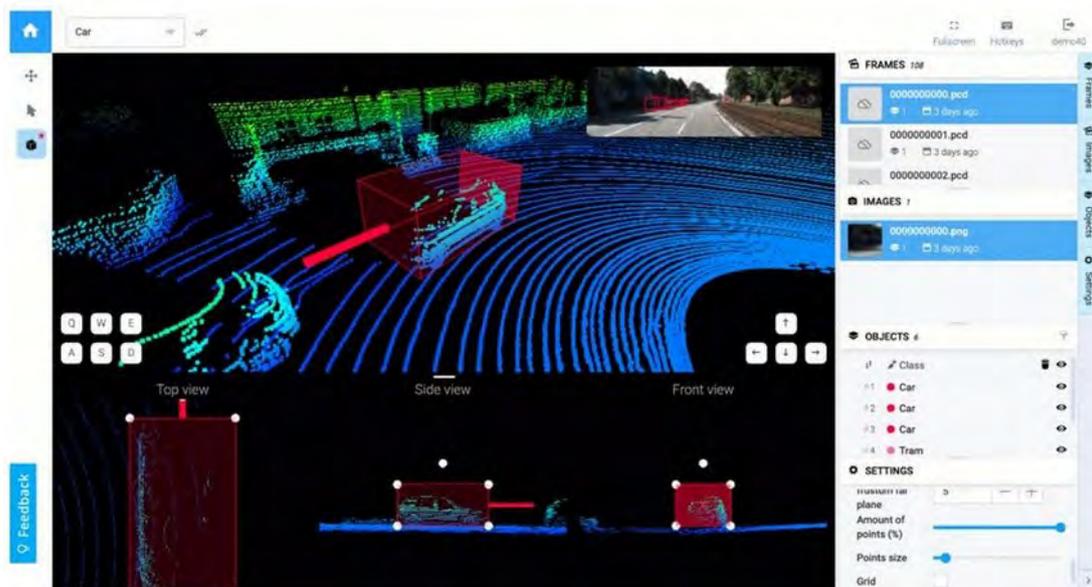
Par rapport aux outils SIG la visualisation des données dans les outils de labellisation images reste assez basique et peut alors se révéler insuffisante dans le cas des données multi-spectrales ou multi-temporelles. Les fonctionnalités de visualisation des objets/vecteurs est aussi assez simple et permet de jouer sur la transparence, une définition de la couleur dans la nomenclature et une sélection/affichage par objet ou groupe d'objet pour les tâches de détection d'objets ou instance segmentation.

La plupart des outils ne supporte pas plusieurs visualisations image comme alterner entre visualisation RVB ou IRC. Et aucun outil n'offre des fonctionnalités permettant de saisir facilement les changements entre dates, par exemple en permettant de visualiser plusieurs images correspondant à plusieurs dates sur une position donnée. Une des seules plateformes commerciales avec une telle option est labelbox qui permet de choisir entre plusieurs layer/overlay en affichage et de passer de l'un à l'autre avec des raccourcis clavier.



Exemple d'affichage avec multiple couches (ici RVB/IRC) avec [labelbox](#)

Les outils de labellisation avec les fonctionnalités de visualisation les plus poussées sont ceux spécialisés dans l'annotation de données pour les véhicules autonomes, avec annotation multi-modale image et lidar ou bien les outils spécialisés pour une thématique précise comme les images médicales. Par exemple dans le cas de saisie 3D sur les données lidar des outils proposent une saisie multi-vues (comme les logiciels de dessin CAO) afin de mieux visualiser les limites des objets en 3D.



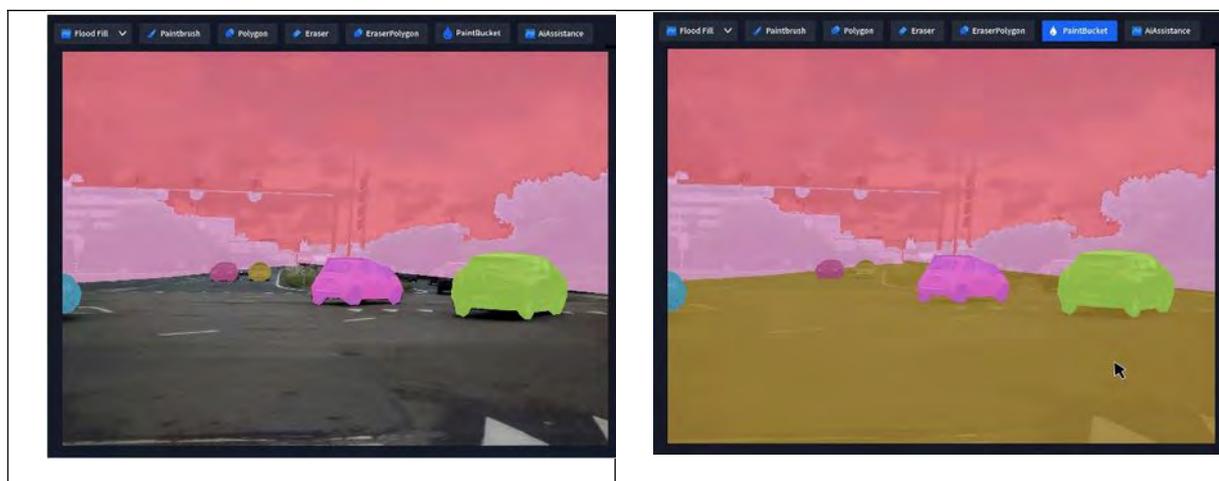
Outils de labellisation de nuage de points (© supervisely)

Saisie manuelle "simple"

La saisie manuelle est le niveau zéro d'automatisation. Il faut distinguer deux familles d'outils correspondant à **deux modes d'édition : image ou vecteur**. Dans le premiers cas les outils s'apparentent à ceux des logiciels d'éditations/retouches images comme Gimp ou Paint. Dans le second cas les outils sont plutôt semblables à ceux que l'on retrouve dans les outils SIG ou DAO.

Pour l'édition de type raster/image on trouve selon les logiciels et plateformes les outils suivants :

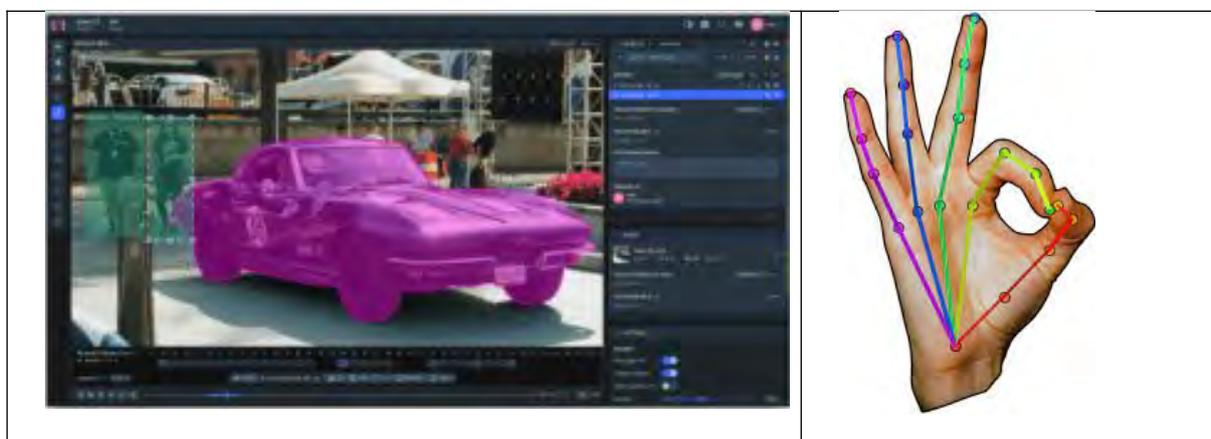
- **Pinceau** (brush) : coloriage/affectation de pixel à une catégorie. Souvent il est possible de configurer la taille du pinceau (en taille pixel).
- **Gomme** (eraser) : suppression de label de pixels. Equivalent au pinceau.
- **Polygone** / boite : le dessin de la forme est fait en mode vecteur mais une fois la saisie terminée la forme est rasterisée sur l'image à saisir et ne peut être modifiée que par les outils raster et non par les outils vecteurs (déplacement de sommets)
- **Pot de peinture** (bucket) : remplis tous les pixels non affectés d'une zone donnée avec une couleur/classe.



Exemple de saisie avec pot de peinture (© playment.ai)

Au niveau du mode de saisie vecteur on relève les outils suivants :

- **Points** : Pour la saisie de ponctuels (par exemple des yeux) et points d'intérêts.
- **Polyline** : Pour la saisie de limites ou bien d'objets de type linéaires comme des marquages routiers, fils électriques.
- **Polygon** : Pour la saisie d'objet de type surfacique. Par exemple bâtiment, arbres. Il s'agit du mode de saisie le plus courant en segmentation sémantique.
- **Bounding box et Rotated box** : Délimitation d'un objet par une boîte englobante. Principal outil des logiciels pour les tâches de détection d'objet.
- **Ellipse/cercle** : Variante de la boîte englobante pour des objets aux formes plus spécifiques (arbres, œufs, roues ...)
- **Template** : exemple de label pour des tâches d'estimation de pose (personne, main). Ces templates sont de type graph (points et lignes).
- **Cuboid** : pour la saisie d'objets 3D, principalement personnes/véhicules dans le cadre des données de conduite autonome.



Gauche interface [supervisely](#) (©), droite template main (©T Simon, <https://arxiv.org/pdf/1704.07809.pdf>)

Afin de rendre la saisie plus ergonomique les logiciels proposent souvent des raccourcis clavier pour passer entre édition/suppression (brush/eraser) ou pour annuler les derniers points et objets saisis (**Undo/redo**). Il n'y a par contre pas d'outils d'édition de type fusion/découpe de polygones comme le propose ArcGIS et [QGis](#) :

“You can split features using the  Split Features icon on the toolbar. Just draw a line across the feature you want to split

The  Merge Selected Features tool allows you to create a new feature by merging existing ones” (© QGis)

Saisie assistée

Les outils de saisie manuelle standards peuvent rapidement se révéler chronophages, en particulier pour les tâches de segmentation et saisie de polygones. Un ensemble d'outils d'édition assistée peut alors être proposé pour rendre la saisie plus efficace. Ces outils utilisent des méthodologies de traitement

d'images relativement simples et représentent le niveau 1 d'automatisation de la saisie.

Magic wand / pipette magique :

Outils provenant des logiciels d'édition image/retouche photo. Sélection d'une surface à partir des valeurs d'un pixel (clics souris) et d'un seuil de distance à la "couleur" du pixel. Plus performant sur les images avec des textures homogènes (aplats de couleur) que sur les images avec des textures complexes.



Saisie avec [pipette magique dans hardy.ai](#) (©). Le cercle jaune représente les pixels sélectionnés manuellement (clic+ taille rayon). La surface orange est la sélection alors obtenue avec la méthode pipette magique.

Contour aka magic-lasso / edge-seeker:

Accroche les sommets d'un polygone sur le contour de l'image le plus proche. Permet d'avoir une saisie à main levée et moins précise donc plus rapide. Pertinent pour les objets avec fort contraste par rapport à l'arrière-plan.

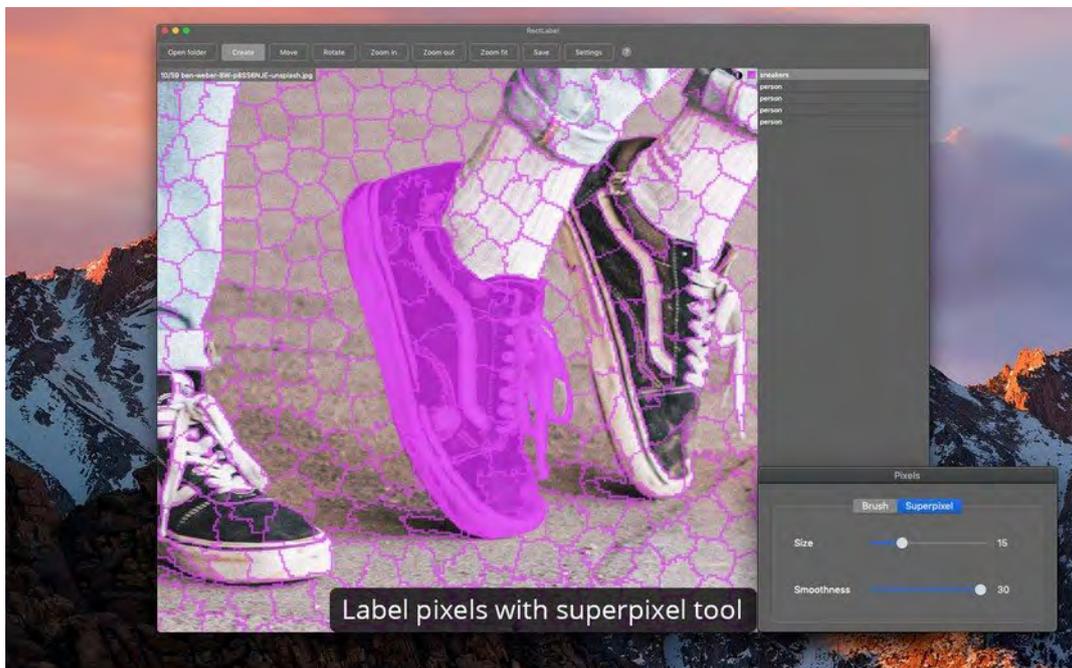


Exemple de saisie avec suivi de contour ([krita](#) ©). Droite, en mode main levée un sommet est ajouté selon un critère de distance fixe.

Segmentation / superpixel :

L'annotateur est assisté dans la tâche de segmentation sémantique par un calcul préalable de segmentation image, généralement de type superpixel/SLIC et donc avec des segments assez compacts. La tâche de labellisation est alors transformée de la saisie de polygone en la sélection de segments/polygones

appartenant à une classe objet. Ce changement de tâche permet déjà de gagner du temps mais afin que cela soit encore plus efficace la segmentation est souvent accompagnée d'outils de sélection avancés comme la sélection par lasso ou polygone ou le fait de pouvoir faire des sélections à main levée (sans cliquer sur chaque segment).



Saisie superpixel avec [RectLabel](#) ©

Afin que ce type d'outils soit efficace il est aussi important de **choisir un algorithme de segmentation adapté aux données à annoter**. Certains algorithmes récents offrant de meilleures performances sur des images extérieures que certains algorithmes de superpixel très répandus comme SLIC. De plus tous les objets à saisir n'étant pas forcément à la même échelle, le fait de pouvoir éditer des **segmentations de façon hiérarchique** (sur plusieurs niveaux de précisions) peut aussi faciliter la saisie.

Saisie avec automatisation

Afin de rendre la saisie encore plus efficace des outils encore plus automatisés ont été mis en place. Il s'agit premièrement des outils d'automatisation de niveau 2 qui changent la tâche de saisie opérateur par une tâche plus simple et calculent de façon automatique la saisie demandée initialement à partir "d'indices" saisis manuellement.

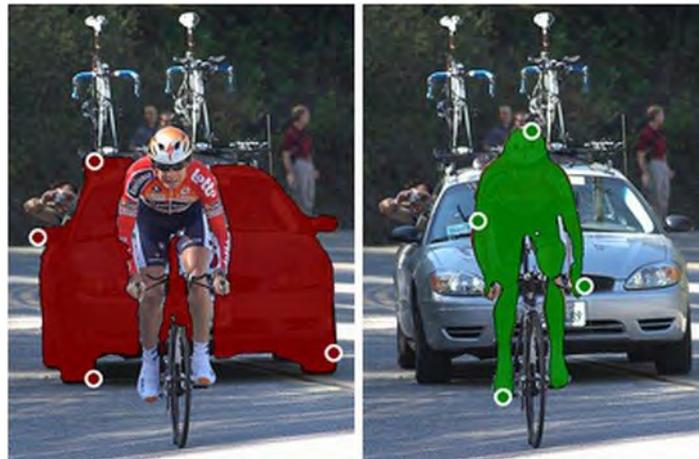
Les interactions utilisateurs possibles pour saisir les "indices" sont alors :

- **Points** : l'utilisateur place/indique des points positifs et négatifs par rapport à l'objet. A chaque ajout de point un masque est mis à jour de façon interactive et l'utilisateur peut ajouter d'autres points sur les parties mal estimées jusqu'à convergence de l'algorithme.
- **Bounding boxes** : l'utilisateur saisit une boîte englobante autour de l'objet et un algorithme calcule ensuite une classification binaire (foreground/background) dans cette enveloppe.

- **Scribbles (gribouillages)** : l'utilisateur dessine des traits/marques sur les différentes classes de l'images et l'algorithme en déduit un masque de segmentation.

Cette famille d'outils de niveau 2 comprend entre autres :

- **GrabCut** : algorithme d'extraction arrière/avant plan d'objet à partir d'une saisie de boîte englobante ou bien d'indices sur les parties intérieures/extérieures à l'objet.
- **DEXTR** modèle deep learning qui prend en entrée une image et un masque avec quatre points représentant les points extrêmes d'un objet (les points les plus haut/bas/à gauche/à droites) et prédit un masque de l'objet par rapport à son environnement.



Exemple utilisation [Dextr](#) (© K.K. Maninis)

- **Inside/outside guidance** : Clic au milieu de l'objet à segmenter. En fonction du masque calculé et affiché on peut ensuite ajouter des points positifs ou négatifs. Pour des objets simples cela permet de faire des annotations en 2 à 5 clics. Cela demande un peu d'expérience avec l'outils afin d'être capable de prédire un peu le résultat de l'algorithme et savoir où il est préférable de placer les points pour mieux guider l'algorithme.



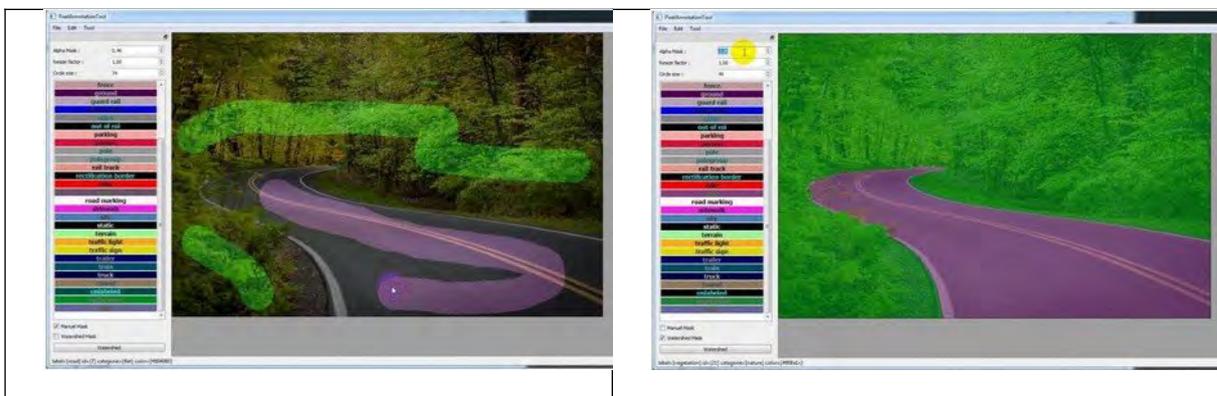
Exemple de segmentation avec inside/outside guidance. Outil [ATOM Segmenter](#) de Hasty.ai ©

La plupart de ces méthodologies d'automatisation concernent la segmentation d'objet et comptent donc implicitement sur une séparation avant/arrière-plan. Dans le cas des données d'occupation du sol ou autres données images

géomatiques ce n'est pas toujours un pré-requis que l'on peut avoir car on ne saisit pas toujours des objets avec des limites nettes et **la notion de sujet principal et avant/arrière plan est moins prégnante dans le cas des images aériennes et spatiales.**

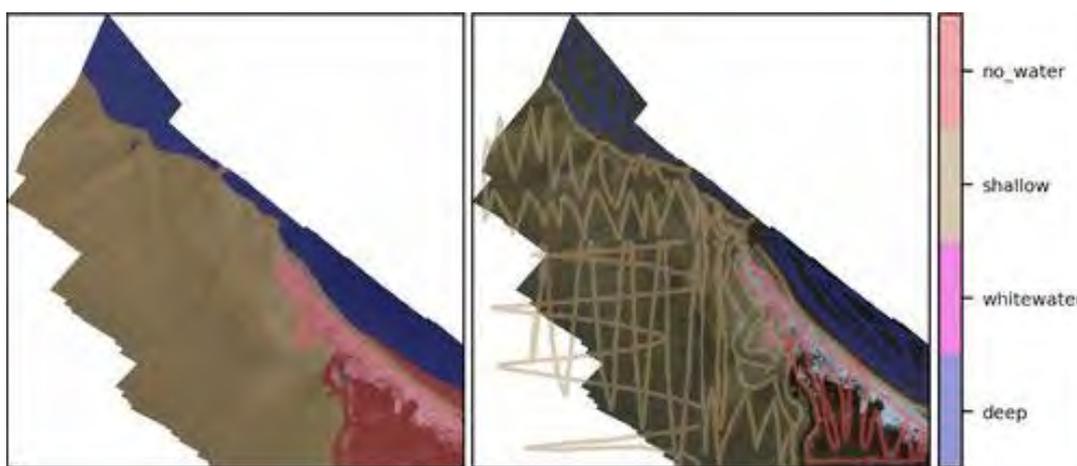
Dans ces cas où la notion d'objets/sujets est moins pertinente Il existe d'autre famille d'outils qui cherchent plutôt à obtenir une segmentation (classification au pixel) de l'ensemble de l'image :

- **Masked watershed** : notamment disponible dans l'outils [PixelAnnotationTool](#). Une segmentation de type watershed³ est d'abord effectuée sur l'image. Les différentes régions sont alors obtenues à partir des marques de l'utilisateur et d'un algorithme de séparation de régions par "montée des eaux". Le résultat est assez dépendant de la bonne segmentation de l'image et de la façon dont le calcul de la segmentation watershed est faite.



Exemple de segmentation avec pixelannotationTool (©)

- Classification / CRF model : classification supervisée au pixel à parti d'échantillons saisies par l'utilisateur. Par exemple en géomatique l'outils [dl_doodler](#) utilise une classification à partir d'une méthodologie CRF (conditional random field)



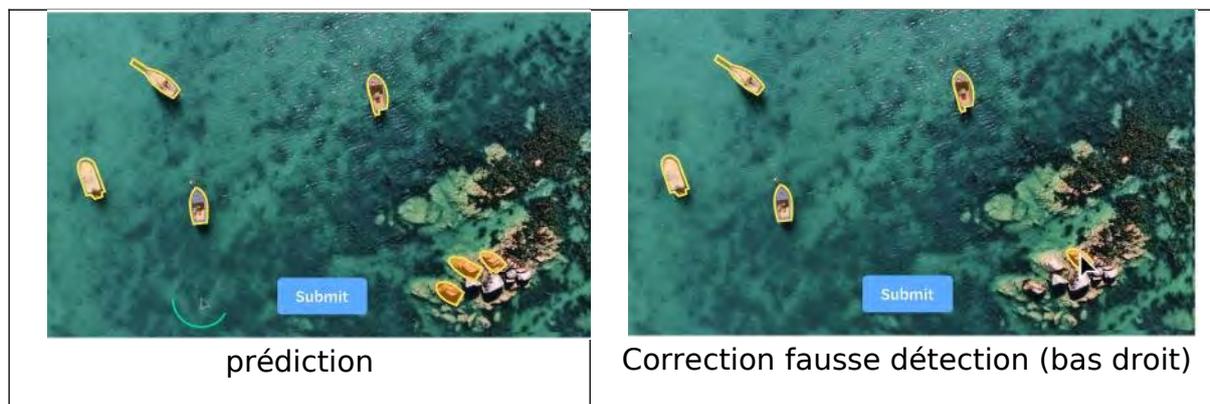
Exemple de saisie avec l'outil dl_doodler. (© Daniel Buscombe)

³ watershed/segmentation par "montée des eaux". Algorithme issu des traitements d'images en morphologie mathématique.

Dans le cas d'occupation du sol certaines plateformes de saisie collaborative sont basées sur ce type de méthodologie, par exemple la plateforme [PEARL](https://developmentseed.org/blog/2021-05-17-pearl-ai-land-cover) (travaux conjoint Microsoft AI for earth et development seed) propose de calculer une occupation du sol de façon interactive par dalle. (<https://developmentseed.org/blog/2021-05-17-pearl-ai-land-cover>)

Dans le cas de la détection d'objet, une fois le masque de segmentation calculé, un modèle de classification image peut alors être appelé pour faire des propositions automatiques de classe que l'opérateur peut valider ou corriger.

Le niveau d'automatisation suivant consiste à **calculer des pré-annotations** de classification/détection ou segmentation et de demander à l'opérateur de faire des corrections si besoin et sinon de juste valider le résultat. **Ce type d'automatisation n'est rentable que si on dispose d'un modèle de pré-annotation déjà relativement fiable** et si les corrections manuelles ne sont pas chronophages. Par exemple il est préférable d'avoir un modèle qui sur-détecte des objets mais trouve les limites de façon correcte plutôt qu'avoir un modèle avec une bonne détection d'objet mais des résultats de segmentation moyens car la suppression de fausse détection/objet est plus rapide que la correction de mauvaises limites objets.



Exemple de correction d'annotation automatique ([labelbox](#))

Pour ce dernier degré d'automatisation il est alors important de prédire aussi **une estimation fiable de la confiance en la prédiction**. Cela permet d'une part de sélectionner les images où le modèle rencontre des difficultés et de planifier des labellisations "classiques" sur celles-ci et d'autre part de ne soumettre à validation/correction que des images où cela doit être plus rapide qu'une annotation à partir de zéro.

Analyse des outils existants

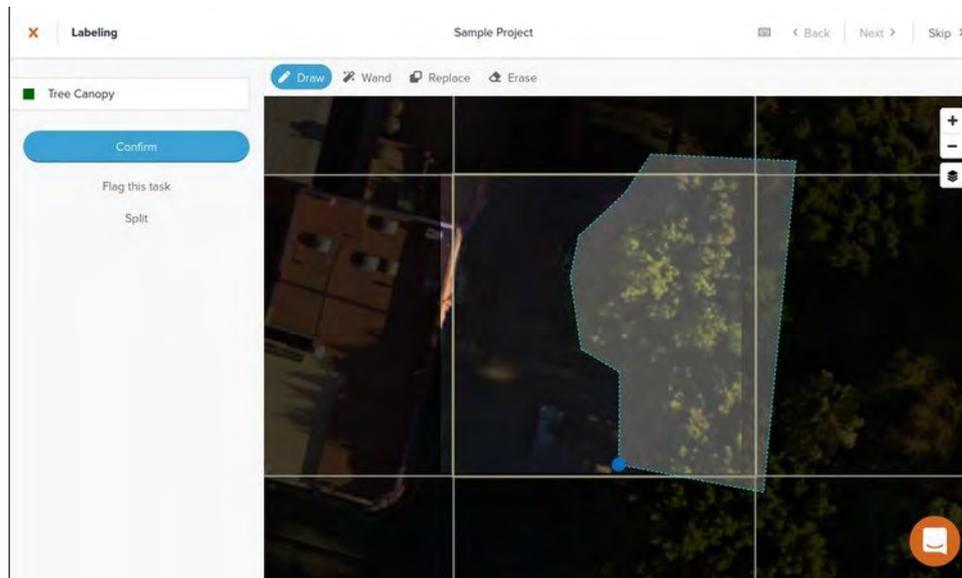
On propose ici une revue rapide des outils existants au vu des besoins IGN dans les trois grandes familles, outils propriétaires, open-source et outils géomatiques

Outils SIG & Télédétection

Les outils SIG proposant des fonctionnalités de labellisation de données d'apprentissage sont assez peu nombreux et peuvent donc être revus de façon presque exhaustive. Nous n'abordons toutefois pas ici les outils générant des labels de façon automatique par rasterisation de base de données géographiques existantes. A l'inverse nous présentons quelques outils qui ne sont pas conçus pour la labellisation de données d'apprentissage mais proposent des fonctionnalités de saisie automatisée pour données géomatiques.

Groundworks

([lien](#)) Outil de labellisation de données géographiques proposé par la société Azavea. Parmi les outils SIG il est celui qui se rapproche le plus des outils de labellisation de données IA. Il est par contre assez limité en termes de fonctionnalités par rapport à ces derniers. D'une part les **outils de saisies de données sont basiques et ne proposent aucune automatisation** et d'autre part les fonctionnalités de gestion de projet et de la donnée sont également peu développées.

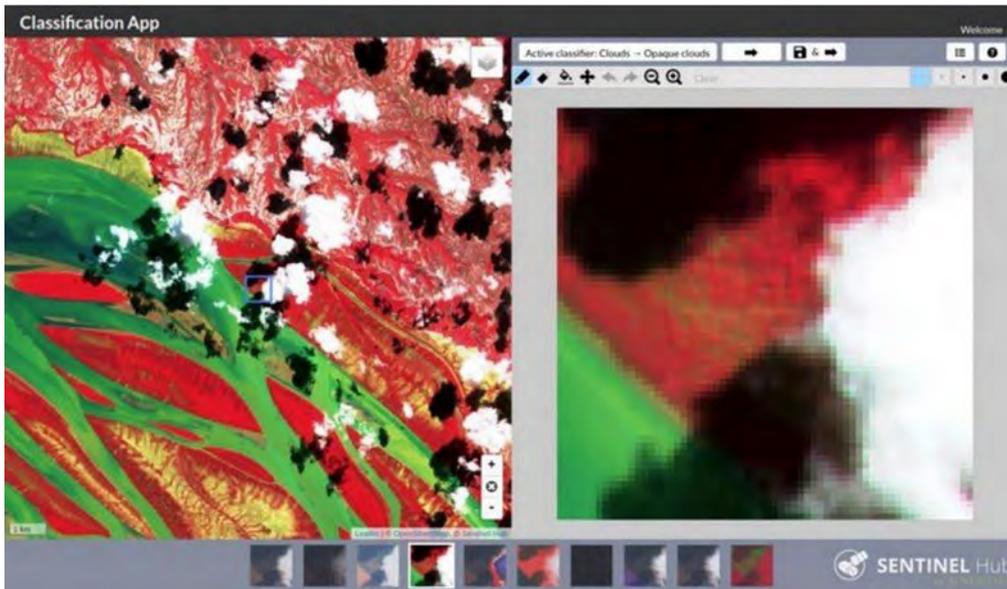


Interface de saisie de l'outils Groundworks (©Azavea)

Cet outil est pensé pour l'accès aux données images sur les technologies STAC (catalogue image) + COG (cloud optimized geotiff)

Sentinel-hub classification app

[Application web](#) proposée par la société Sinergise et développée autour des services de diffusion et visualisation de données Sentinel que celle-ci propose. L'application a été développée initialement pour la création d'un jeu de données d'apprentissage pour la détection de nuages (nuages denses, fins et ombres) sur les données sentinel-2. Les outils de saisie sont très simples et seulement raster (pinceau, gomme, pot de peinture). L'intérêt majeur de l'outil est de disposer des fonctionnalités de visualisation de données multi-spectrales proposées par le service Sentinel-hub de Sinergise. Celui-ci permet d'afficher l'image selon plusieurs compositions colorées.

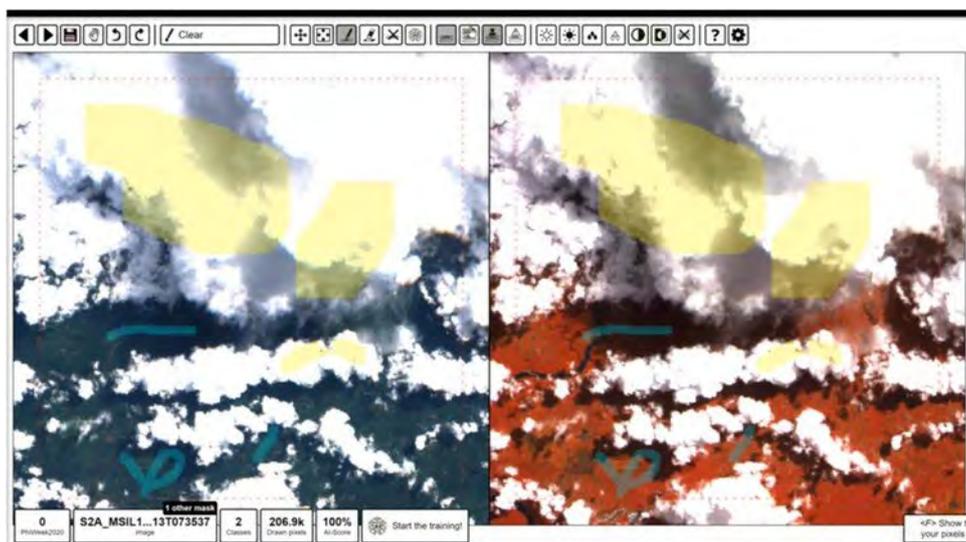


Interface de saisie de l'outil Classification-App (© Sinergise)

Le code côté client est disponible en [open-source](#)

IRIS

([lien](#)) Outil open-source développé pour la constitution d'un jeu de données d'apprentissage pour la segmentation de nuages sur les données satellites (ici sentinel-2). La visualisation propose plusieurs fenêtres synchronisées permettant d'afficher plusieurs vues/compositions colorées différentes. La labellisation est faite par saisie d'échantillons sur des exemples puis classification (de type lightGDM / random forest) sur le reste de l'image.

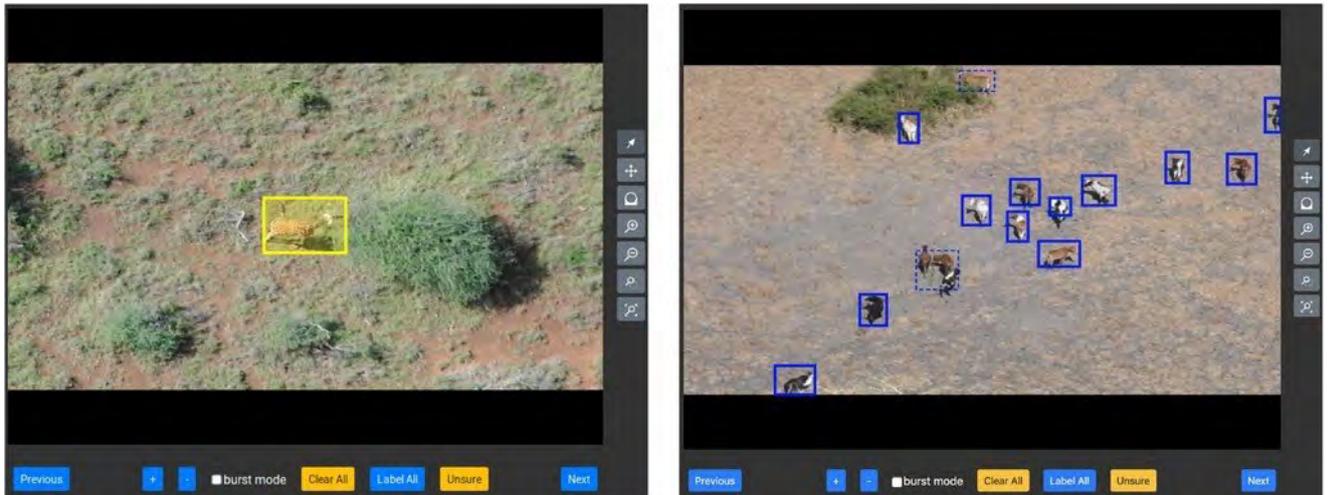


Exemple de saisie des labels IRIS avant classification (© Esa Phi-lab)

AIDE

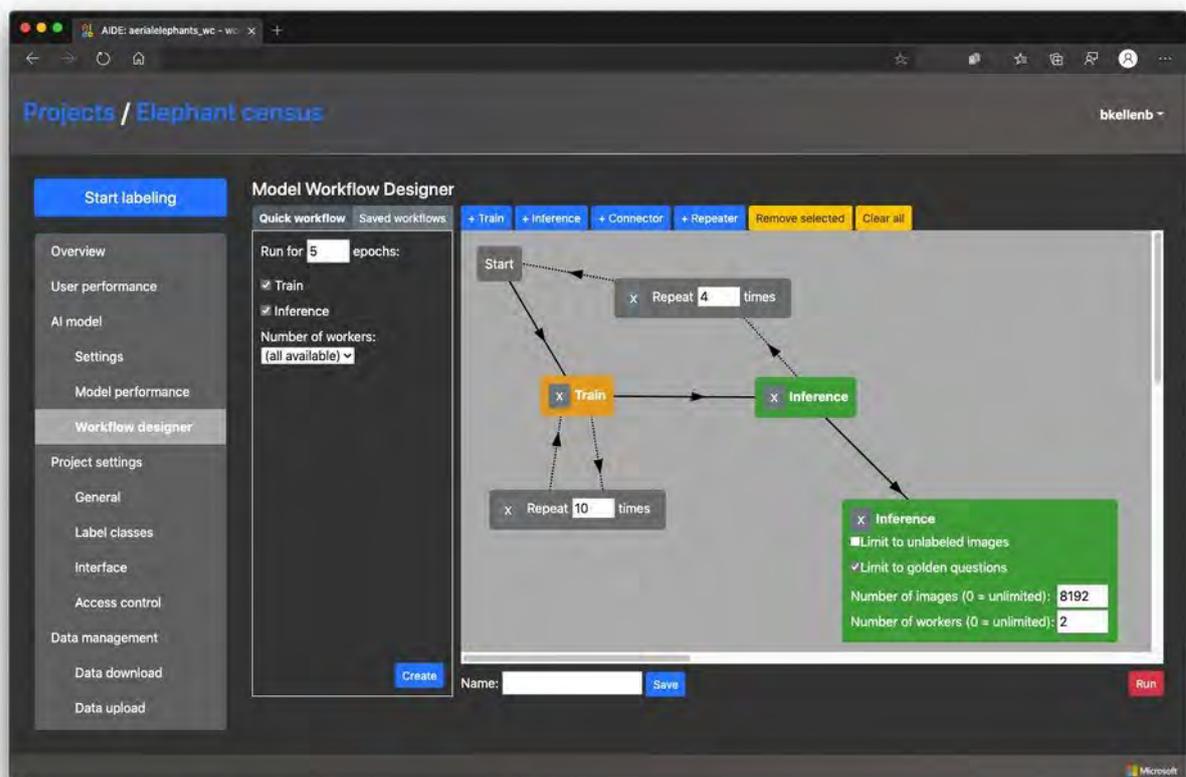
([lien](#)) Annotation d'image aérienne type drone ou très haute résolution pour extraction d'animaux (ou éléments importants pour l'écologie). Le logiciel est open-source et supporté par Microsoft dans le cadre du programme 'Microsoft AI for Earth' , il peut être déployé sur les offres cloud Microsoft Azure.

Un des points forts de l'outil est de proposer une méthodologie d'annotation de type active learning et propose des modèles de détection pré-entraînés sur étagère. Ceux-ci peuvent ensuite être adaptés/spécialisés à une problématique donnée par l'utilisateur via la saisie d'échantillons supplémentaires.



Exemple d'interface de visualisation et saisie de AIDE (© B. Kellenberger, Microsoft)

De plus l'outil propose quelques fonctionnalités de gestion de projet et de données. Il offre en particulier la possibilité de configurer le workflow d'active learning par projet en permettant de paramétrer à quelle fréquence relancer des entraînements/spécialisations du modèle de détection utilisé. Plus d'information sont disponibles sur [l'article](#) associé à l'outil.



Configuration du workflow d'active learning d'un projet AIDE

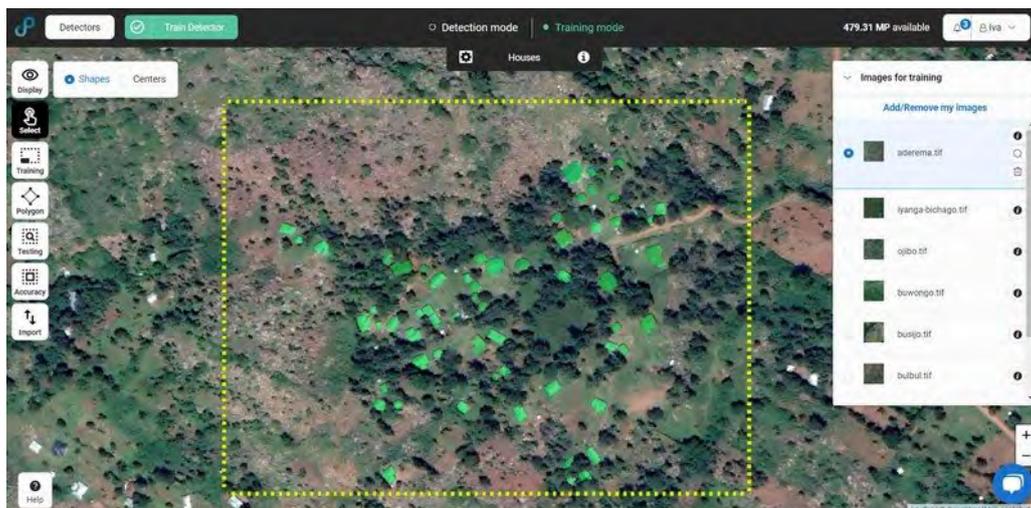
Esri Arcgis Pro

Depuis quelques années ESRI essaye de développer des fonctionnalités d'analyse deep learning dans sa suite [Arcgis-Pro](#). Les fonctionnalités sont relativement peu poussées et consistent à utiliser les outils d'éditations SIG déjà existants, en particulier les outils vecteurs pour la saisie de boîtes englobantes ou de boîtes englobantes orientées. Il est également possible de calculer une segmentation sur les images et de labelliser ces segments. Il ne s'agit par contre pas d'une segmentation hiérarchique.

Il n'y a pas non plus de fonctionnalités d'automatisation de saisie plus avancé comme grabcut/Dextr ou bien des fonctionnalités de gestion de projet, de données ou d'éditations collaborative associées. Il est par contre possible de faire des prédictions à partir d'un modèle entraîné. Esri propose d'ailleurs des modèles déjà entraînés sur certains thèmes.

Picterra

La société [picterra](#) propose une plateforme permettant de produire des détections d'objets géographiques par une méthodologie d'active learning. Le **but n'est pas alors de créer un jeu de données d'apprentissage mais d'entraîner directement un "détecteur" sur un certain type d'objet** et de données. L'utilisateur charge sur la plateforme ses données images (drone, aérienne ou spatiale) et effectue une saisie de quelques éléments. Une première détection est alors proposée par Picterra sur une zone définie par l'utilisateur et le modèle peut être amélioré de façon itérative.



Interface de l'outil Picterra (©)

L'utilisateur peut à la fin récupérer les résultats de détection mais non pas les données d'apprentissage ou bien le modèle de détection. Afin de faciliter la création de projet la société propose aussi d'accéder directement aux services de certains fournisseurs de données aériennes ou spatiales.

OSM & MapsWith.ai:

Les outils de l'écosystème OSM ont pu être utilisés tels quels pour la vectorisation collaborative de cartes anciennes. Concernant une automatisation plus poussée de la saisie et l'utilisation de technologie deep learning il y a l'exemple de l'expérimentation MapsWithai de Facebook qui propose un outil d'aide à la détection/saisie de route à partir de détection deep learning.



Workflow proposé par MapsWithAi (© Facebook)

Il ne s'agit toutefois pas d'un outil de saisie de données d'apprentissage mais plutôt d'un outil d'aide à la saisie de base de données géographiques, la différence entre les deux n'étant pas toujours très grande. L'interface a été développée en modifiant l'éditeur de données OSM "iD"



Exemple de routes candidates après détection deep learning et comparaison avec l'existant dans OSM (© Facebook)

Autres outils SIG/géomatiques :

- [Geohive](#) : expérimentation sur un outil de labellisation de type crowdsourcing à partir de données satellites proposées par la société Maxar. Après de premières publications/communications l'outil ne semble pas évoluer ou être promu de façon active.
- [collect.earth](#) : plateforme de crowdsourcing/partage de données d'apprentissage pour classification de données satellites. Plutôt orienté pour les algorithmes machines learning que deep learning. Mise en place dans le cadre d'une collaboration NASE-FAO et du projet [openForis-Sepal](#) (System for earth observation, data access, processing, analysis for land monitoring). Editions simple de polygone sur une interface web de visualisation de données images spatiales.
- **Qgis / Grass** : Il n'existe pas de fonctionnalité de labellisation de données d'apprentissage connue pour ces SIG open-source.

Outils et plateforme propriétaires

A l'opposé des outils SIG il existe de très nombreuses plateformes et outils de labellisation de données d'apprentissage. Nous nous limitons ici aux outils ayant des fonctionnalités de saisie automatique pour la tâche de segmentation

sémantique ou bien à certains ayant des caractéristiques particulières et intéressantes à noter.

SuperAnnotate

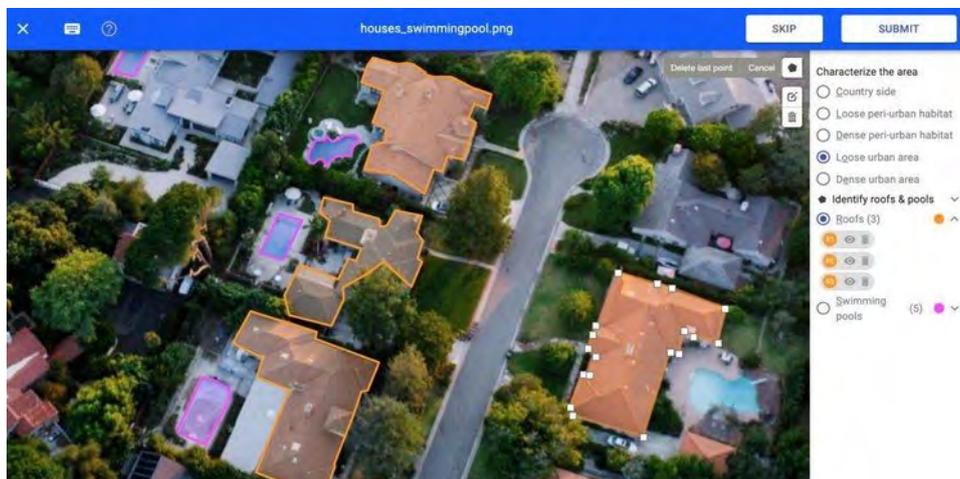
Plateforme de labellisation de données plutôt spécialisée dans l'annotation d'image et de vidéo. Son principal intérêt est le fait de proposer un outil de saisie de données de segmentation sémantique qui semble performant/efficace et relativement compatible avec les données géomatiques. Ce dernier est basé sur une segmentation hiérarchique et propose une interface graphique relativement bien pensée pour cette tâche. Elle propose aussi une version desktop de son outil d'annotation mais sans la partie segmentation qui dépend de calculs sur un serveur, la version plateforme peut être déployée sur une infrastructure propre.



Interface de superannotate pour la saisie de segmentation (© superannotate)

Kili-technology

Plateforme d'annotation, également plutôt orientée image et vidéo. Un premier intérêt est qu'il s'agit d'une des rares plateformes françaises sur le marché, ce qui peut simplifier des choses en cas de marché/apel d'offres. D'autre part elle offre des fonctionnalités assez complètes (gestion projet, gestion qualité, différents degrés d'automatisation possible et entraînement de modèles deep learning). Son outil d'édition pour la tâche de segmentation sémantique semble un peu moins évolué que celui de superannotate mais il continue d'évoluer et propose aussi depuis peu l'utilisation de plusieurs niveaux de segmentation pour faciliter l'annotation (niveau de segmentation superpixel slic).



Interface de Kili-technology pour la saisie de segmentation/instance (© Kili-technology)

LabelBox

Une des plateformes de labellisation les plus présentes et anciennes sur le marché. Elle propose des fonctionnalités plutôt complètes et compatibles avec la plupart des tâches d'annotation possibles. Concernant les données géomatiques la plateforme accepte les données multi-couches (multi-layer) ainsi que des sources de données de type flux d'image WMTS.



Labellisation d'image spatiale avec les outils labelBox (©) utilisant une pré-segmentation image.

Supervise.ly

([lien](#)) Plateforme de labellisation plutôt spécialisée sur le marché des véhicules autonomes. Elle propose des outils de labellisation image et lidar ainsi que vidéo ou séquence d'images. Un de ces points forts concerne la gestion des données labellisées (combinaison de jeux de données, augmentation de jeux de données, pré-traitement) ainsi que l'entraînement de modèles deep learning et l'automatisation de la labellisation. Par contre les fonctionnalités sur le contrôle qualité et les outils d'éditeurs pour la segmentation sémantique sont un peu moins convaincants que les autres plateformes. En particulier les outils de segmentation sont plutôt orientés sur la segmentation objets et non sur la segmentation de toute l'image, les outils proposés ne sont alors pas forcément adaptés pour les images aériennes ou spatiales.

Outils et solution open-source

Label Studio

([lien](#)) Plateforme de labellisation fonctionnant sur un double modèle. Une partie open-source et une offre commerciale propriétaire. La plateforme open-source est plus limitée que la partie commerciale, notamment au niveau de la gestion des droits utilisateurs et des outils de contrôles qualité. Par contre la communauté open-source semble assez active pour le moment. Toutefois il n'y a pas de fonctionnalités de labellisation de segmentation sémantiques basées sur une segmentation image pour l'instant (fonctionnalité [à l'étude](#)). L'intérêt de la plateforme est donc actuellement d'offrir une base de développement assez

robuste et complète qui peut être adaptée pour un besoin propre. La plateforme est actuellement constituée de plusieurs projets relativement indépendants :

- `label-studio` : partie serveur de la plateforme de labellisation, gestion de données.
- [label-studio-frontend](#) : client web de labellisation basé sur React et javascript.
- [Data-manager](#) : client web de gestion de données labellisé utilisant le serveur `label-studio` et des composants du frontend.
- [label-studio-converter](#) : utilitaire/librairies de conversion de format entre données labellisée et outil/framework IA.

Django Labeller

Logiciel de labellisation pouvant fonctionner en mode desktop ou client/serveur. Django vient du nom d'un Framework de développement web python. Parmi les logiciels de labellisation open-source il est un de ceux offrant les fonctionnalités de saisie sur données images les plus complètes. En particulier :

- Saisie de point, polygone, ellipse
- Mode de saisie raster/image avec brosse et gomme. La brosse et la gomme peuvent servir sur les objets vecteurs aussi (passage raster <-> vecteur)
- Possibilité d'automatisation avec des algorithmes de type Dextr et grabCut en mode client/serveur.

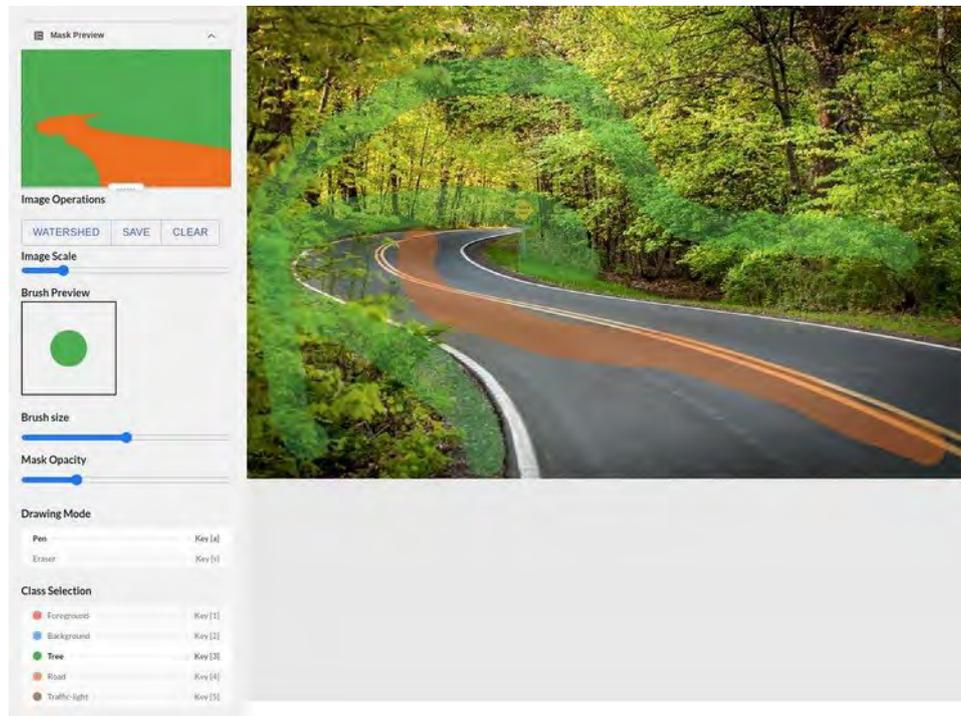


[Utilisation de l'outil brush \(mode erase\) pour editer une segmentation django labeller](#)

Les codes sont en python (partie algorithmique et backend) et interface desktop (avec Framework Qt) et javascript/vue.js pour la partie cliente web.

PixelAnnotationTool

Outils de labellisation image basé sur une segmentation de l'image puis classification des segments à partir des indices/marqueurs saisie par l'utilisateur. La version initiale a été codée en python et utilise les outils de traitement d'image et machine learning existant dans le framework opencv. Une version similaire a été portée sous forme de client web ([pixel-annotation-tool-written-in-react-and-opencv-js](#)). L'algorithme de segmentation utilisé est celui du marked-watershed qui provient des travaux de traitement d'image en morphologie mathématique.



Interface web pour PixelAnnotation Tool (© [Rand Xi](#))

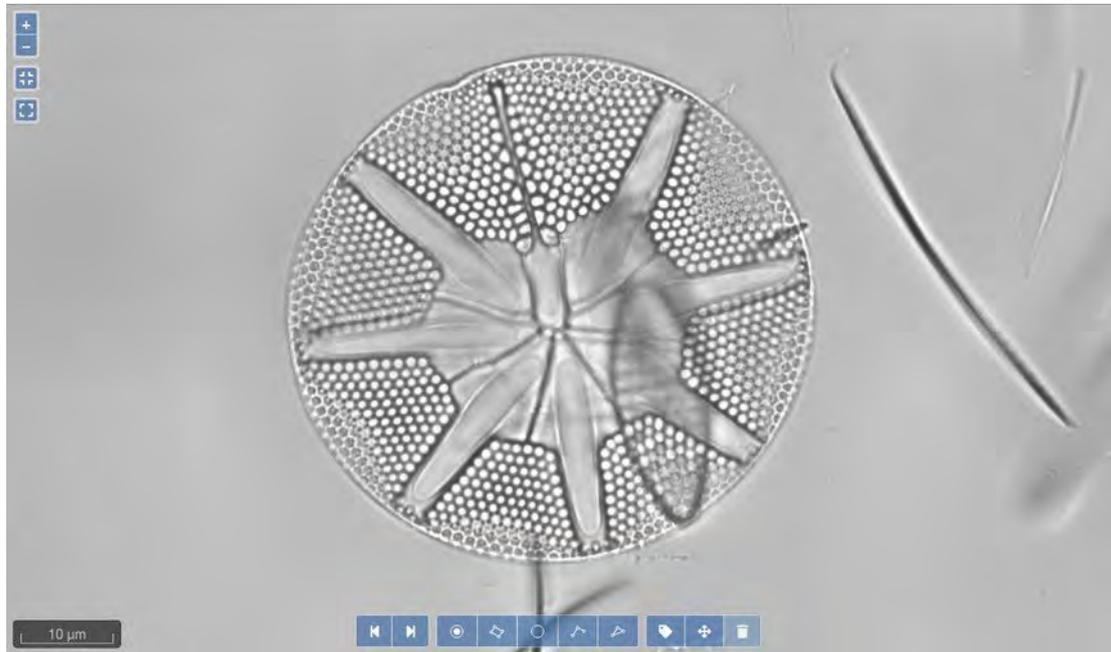
ImageTagger

Plateforme de labellisation open-source dédiée aux compétitions robotiques (match de foot entre robot entre autres). L'intérêt de l'outil est qu'il montre un exemple concret de ce qu'il est possible de développer dans le cadre d'une thématiques/communauté spécialisé et qu'il offre un code dont on peut s'inspirer et étudier les choix techniques. En termes de fonctionnalité cet outil propose :

- Création d'équipe
- Création / gestion de jeux de données image
- Saisie de boîtes englobantes/polygones lignes et points
- Gestion et fonctionnalités d'export de jeux de données labellisé (choix, format d'annotation)
- Possibilité d'utiliser des pré-annotations.
- Des fonctionnalités de contrôle qualité

Biigle ou Viame

[\(lien biigle\)](#) Biigle et Viame sont deux plateformes de labellisation open-source spécialisées dans l'annotation de données images pour l'exploration marine (faune, flore). De façon similaire à ImageTagger ce sont des projets intéressants à étudier en tant qu'exemple de développement open-source de plateforme de labellisation. Mais de plus, comme pour les données géomatiques, ils peuvent être amenés à gérer des images de tailles importantes et dans un contexte différent des prises de vues standards (photo personnelle ou smartphone). Leur gestion des données images est donc plus proches des besoins IGN que les outils d'annotation standard.



Exemple de labellisation avec la plateforme Biigle.

Biigle est un outil de type plateforme web; développé en PHP avec le Framework Laravel côté serveur et angular.js côté client. Il permet de spécifier sa propre nomenclature, de créer des projets avec une gestion des droits utilisateurs, d'explorer et gérer les données annotées et en termes d'édition il offre à la fois les outils d'éditeurs vecteurs classiques et des outils plus automatiques comme la proposition de zones à annoter (présence d'objet) et la proposition de masque de détection (boite englobante) via l'entraînement de modèles deep learning spécialisés.

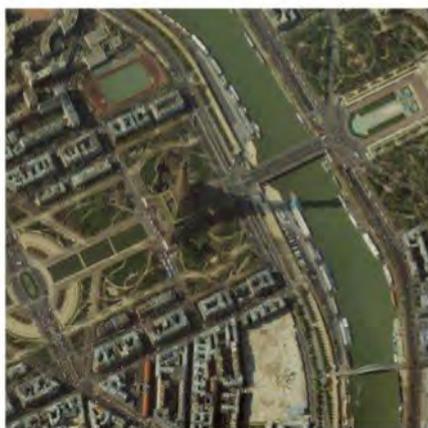
VIAME est un outil développé par Kitware (mainteneur de cmake et de la librairie de traitement d'image ITK qui sert de base à la librairie OTB très utilisée en télédétection). Il n'est pas conçu seulement comme un outil de labellisation mais plutôt comme une plateforme scientifique complète pour l'analyse de données marines (images, vidéo) avec des outils de pré-traitements de données (mosaïquage de données, pré-traitement image), des outils de labellisation et des outils de créations de modèles de détection et leur analyse.

3 Analyse des méthodologies et outils de segmentation pour labellisation image

La majorité des besoins actuels de l'IGN en outils de labellisation concerne des données de type occupation du sol ou cartographie ancienne et sont donc des tâches de segmentation sémantique (ou instances). Parmi les outils les plus prometteurs dans l'étude de l'existant pour cette tâche on peut observer une **majorité d'outils/platformes utilisant une segmentation**, souvent de type superpixel. On peut par exemple citer la plateforme "kili-technology" sur l'intérêt des superpixels :

*"Superpixels are instead really efficient in most use cases. When well calculated only a few clicks allows you to annotate an object within an image. A key functionality that superpixels tools must provide is the ability to change the superpixels granularity in real time. To be really effective this change of granularity must respect the continuity between the boundaries of the different size of superpixels. In addition, the ability to change the color and brightness of those boundaries is real plus when annotating. When the quality of the superpixels is provided and the UX includes the previous functionalities superpixels become a powerful tool to speed up annotation."*⁴

En effet, **la saisie (détourer les éléments de l'image) est la partie la plus fastidieuse de toute la chaîne de traitement** : selon [Lin et al., 2014]⁵, il faut 79 secondes pour détourer un objet avec un outil de sélection de type polygone (22h pour 1000 annotations). **L'utilisation d'une segmentation en pré-traitement de l'image permet donc de gagner en temps d'annotation.** Par exemple d'après les expérimentations de [Caesar et al., 2016]⁶ par rapport à une saisie au pixel (outils raster) la saisie de polygone est 1,5 fois plus rapide et la saisie sur segmentation 2.8 fois rapide. La saisie sur segmentation est donc presque deux fois plus efficace que la saisie de polygones.



Exemple de segmentation - Dawa Derksen, OTB Users Days 2016
Large scale Superpixel segmentation using OTB

4 <https://kili-technology.com/blog/what-is-the-best-segmentation-tool/>

5 T. Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context. arXiv:1405.0312, 2014.

6H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context,

Toutefois ces outils sont principalement utilisés sur des images “classiques” (prises du sol, avec canaux RVB). Il est alors particulièrement intéressant dans le cas des données géomatiques d’étudier plus précisément les algorithmes de segmentation existants afin de voir ceux permettant de mieux répondre au besoin de labellisation des données de type images aériennes ou cartographie ancienne. En particulier, on s’intéressera aux algorithmes de segmentation implémentés dans les logiciels SIG ou dans les bibliothèques python qui peuvent s’interfacer facilement avec les outils déjà existants dans les projets OCSGE ou BDForêt.

Définitions et enjeux

Avant toute chose, il convient de définir ce que l’on entend par segmentation et préciser le vocabulaire employé, celui-ci pouvant souvent prêter à confusion quand il est mal employé. De plus, il faut définir les contraintes/enjeux auxquels la segmentation doit répondre dans le cas des outils de labellisation.

Segmentation

Une segmentation d’image est souvent entendue comme le découpage d’une image en régions distinctes [Guigues et al., 2003].

Une segmentation est une partition de l’image en ensembles de pixels homogènes. D’un point de vue mathématique, une partition est une décomposition d’un ensemble en sous-ensembles deux à deux disjoints, et de réunion égale à l’ensemble de départ.

Il s’agit donc de délimiter dans une image les zones cohérentes en termes de valeurs, qui sont homogènes ou de faible variation en couleur, intensité, texture, ou altitude par exemple, selon l’information que l’on possède, dans un certain rayon spatial.

Or, un objet peut être constitué de parties hétérogènes. La segmentation est donc un processus permettant généralement d’extraire des primitives, des régions homogènes, qui permettront d’identifier les différents éléments d’un même objet pour ensuite le reconstituer. Il faut donc distinguer **sur-segmentation** (diviser un objet en plusieurs segments) et **sous-segmentation** (regroupement de plusieurs caractéristiques en un seul segment).



Illustration de sur/sous segmentation [YE HE 2013]

Apport annotation : une sur-segmentation est préférable pour garantir la précision de la classification.

Superpixels

Afin d'éclaircir le propos à venir, nous présentons d'emblée les **superpixels** qui sont des algorithmes largement abordés dans la littérature et utilisés en traitement de l'image pour accélérer les processus de calcul. Leurs propriétés sont détaillées en annexe C.

Superpixels : algorithme de sur-segmentation qui produit des segments locaux et cohérents qui préservent la majeure partie de la structure à l'échelle voulue [[Ren et Malik, 2003](#)]. Les superpixels sont donc homogènes en termes de taille, de forme et de couleur.



Scikit-image - SLIC : Simple Linear Iterative Clustering (©IGN)

Dans cette étude, certains des algorithmes de segmentation présentés sont des algorithmes superpixels. En télédétection, les superpixels sont surtout utilisés en pré-traitement d'une segmentation/classification ultérieure pour remplacer la structure rigide de la grille de pixels. Leurs propriétés leur permettent d'être rapides à calculer et d'apporter un gain de temps dans la suite des calculs par rapport à un traitement au pixel.

Apport annotation : accélérer les calculs, simplicité de prise en main (peu de paramètres)

Segmentation hiérarchique

Dans un contexte cartographique et de production de l'OCSGE, on constate qu'il n'y a pas d'échelle de visualisation unique qui permette de décrire tous les éléments cartographiques d'une orthophotographie du fait de la grande diversité en termes de taille, texture, etc. des objets. C'est pourquoi l'on s'intéresse aux algorithmes de segmentation multi-échelle qui permettent de segmenter les objets de l'image à plusieurs niveaux.

Une segmentation hiérarchique consiste en un « système d'ensembles deux à deux disjoints ou emboîtés » [Guigues et al., 2003]. H est une hiérarchie de l'image I si et seulement si :

- L'ensemble E des régions de l'image I appartiennent à H
- Si deux régions, A et B, appartiennent à H, alors elles sont soit disjointes, soit incluses l'une dans l'autre
- Toute région est la réunion des régions qui sont incluses en elle

Il s'agit donc d'un ensemble de segmentations réalisées à **différents niveaux de détails** de l'image qui implique que les segmentations s'emboîtent, c'est-à-dire que celles de niveau plus grossier puissent être des fusions de régions de niveau de détails plus fins. Cela permet à l'annotateur de gagner en **précision** et en détails à mesure que l'on descend d'un niveau dans la hiérarchie et de voir apparaître de nouvelles formes pertinentes. Cette méthode laisse libre cours à l'annotateur dans la labellisation de données en lui permettant de rechercher les formes qui l'intéressent le plus au sein de la structure multi-échelle. Un autre avantage de ce type d'algorithme est sa **robustesse au bruit**, du fait de la propagation des contraintes de bas en haut de la hiérarchie.



[Exemple de segmentation hiérarchique](#). Les niveaux de gris représentent l'échelle à laquelle le contour apparaît (© Silvio Jamil Ferzoli Guimarães)

Dans cette étude, nous aborderons aussi bien les algorithmes classiques de segmentation que les algorithmes plus poussés de segmentation hiérarchique afin d'avoir un état de l'art plus complet des solutions existantes.

Apport annotation : passer d'un niveau de détails à un autre selon la taille de l'objet à annoter.

Enjeux

La segmentation automatique obtenue par algorithme de type machine learning (standards/non deep learning) est rarement vue comme en but en soi. En effet les recherches passées ont montré que ces techniques n'arrivaient pas à produire des segmentations images équivalentes à celles saisies manuellement

et souhaitées en résultat final. Ce type d'algorithme ne peut, par exemple, pas produire directement une segmentation de type BDForêt ou OCSGE. Par contre ces méthodes se sont révélées très utiles et performantes en tant que traitements d'image en amont d'autres algorithmes. Cela est particulièrement vrai dans les cas des superpixels et de la segmentation hiérarchique où cette opération est essentiellement considérée comme un pré-traitement qui extrait des informations pour simplifier une classification ou accélérer d'autres traitements d'image. Elle doit donc répondre à des **enjeux de qualité, de précision et de performances** : fournir des segments qui conviennent à la nomenclature définie et aux spécifications de taille de segments, en un temps de calcul raisonnable.

Ces enjeux pouvant être contradictoires il s'agit de faire des compromis entre eux et choisir la méthode correspondant le plus à un besoin spécifique. Dans cette étude l'enjeu majeur est le **passage à l'échelle** du processus d'annotation de données sur plusieurs points :

- **L'accessibilité** de l'algorithme
- Le **paramétrage** de l'algorithme pour segmenter correctement les divers types de sol. Enjeu d'exhaustivité, de gestion des classes rares
- La **stabilité** de l'algorithme
- La **taille** de l'ensemble du jeu de données à segmenter
- **L'automatisation** de l'analyse d'images

Ces questions sont approfondies dans la partie suivante.

Outils de segmentation image et réponses aux enjeux et besoins IGN

Cette partie vise à mettre en avant les performances, avantages et inconvénients des algorithmes de segmentation disponibles dans les logiciels SIG (type QGIS et eCognition), ainsi que ceux accessibles en programmation par la librairie Scikit-image par rapport aux enjeux définis ci-dessus. La liste des solutions étudiées n'est pas exhaustive mais couvre l'essentiel des outils utilisés en géomatique et télédétection.

Disponibilité des algorithmes

L'algorithme de segmentation doit être accessible facilement pour que son utilisation au sein d'une chaîne de traitement soit fluide, facile à mettre en œuvre et rapide. Selon le besoin, on peut alors préférer un algorithme accessible en ligne de commande, en appelant une bibliothèque, ou à travers un SIG.

Type de segmentation	Accessibilité	
	SIG	Code
Felzenszwalb	RSGISlib	Scikit-image
Quickshift	RSGISlib	Scikit-image
SLIC	RSGISlib , module remote OTB-Slic , eCognition , GRASS addon i.superpixels.slic	Scikit-image

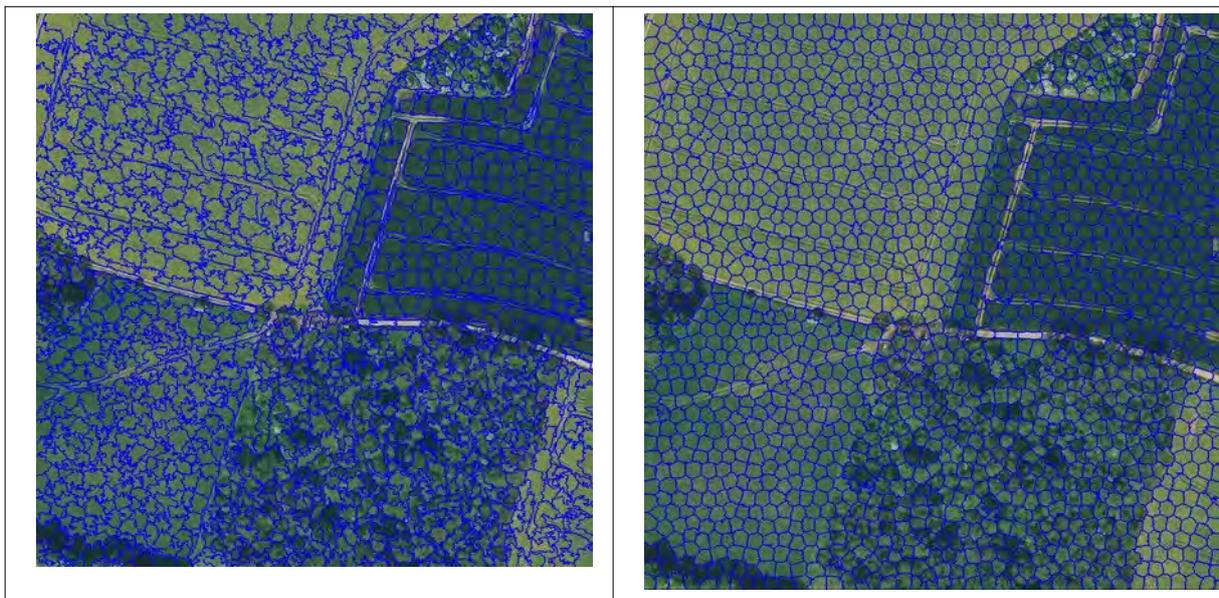
Watershed	RSGISlib , SAGA , GRASS	Scikit-image
MeanShift	QGIS , ArcGIS	Ligne de commande , Python (otbApplication)
LargeScaleMeanShift (OTB)	QGIS	Ligne de commande , Python (otbApplication)
Multi-échelle	ECognition, Pyram	

Implémentations des différents types d'algorithmes

RSGISlib est un logiciel open source intitulé Remote Sensing and GIS Software Library. ECognition est un logiciel d'analyse d'images orienté objet qui compte parmi les premiers outils commercialisés à implémenter des algorithmes de recherche qui exploitent des informations de texture et de forme des objets en plus de la radiométrie pour réaliser des segmentations ou des classifications⁷.

Le logiciel eCognition propose 3 algorithmes superpixels, présentés ci-dessous sur la même ortho-image avec les paramètres suivants :

- Nombre minimal d'éléments : 25
- Taille minimale des éléments : 25
- Nombre d'itérations : 10



Algorithme SLIC (©IGN)

Algorithme SLICO (©IGN)

SLICO est la version algorithmique sans paramètres de SLIC qui définit le paramètre de compacité différemment selon la texture des zones afin d'avoir des superpixels de formes régulières dans toutes les régions. C'est donc la régularité dans la forme qui prime sur l'occupation du sol. Les temps de calcul des deux algorithmes sont équivalents⁸.

Le logiciel eCognition étant payant, il n'a pas été possible de faire des comparaisons poussées entre les algorithmes, notamment entre les deux

⁷ infoterre.brgm.fr/rapports/RP-52120-FR.pdf

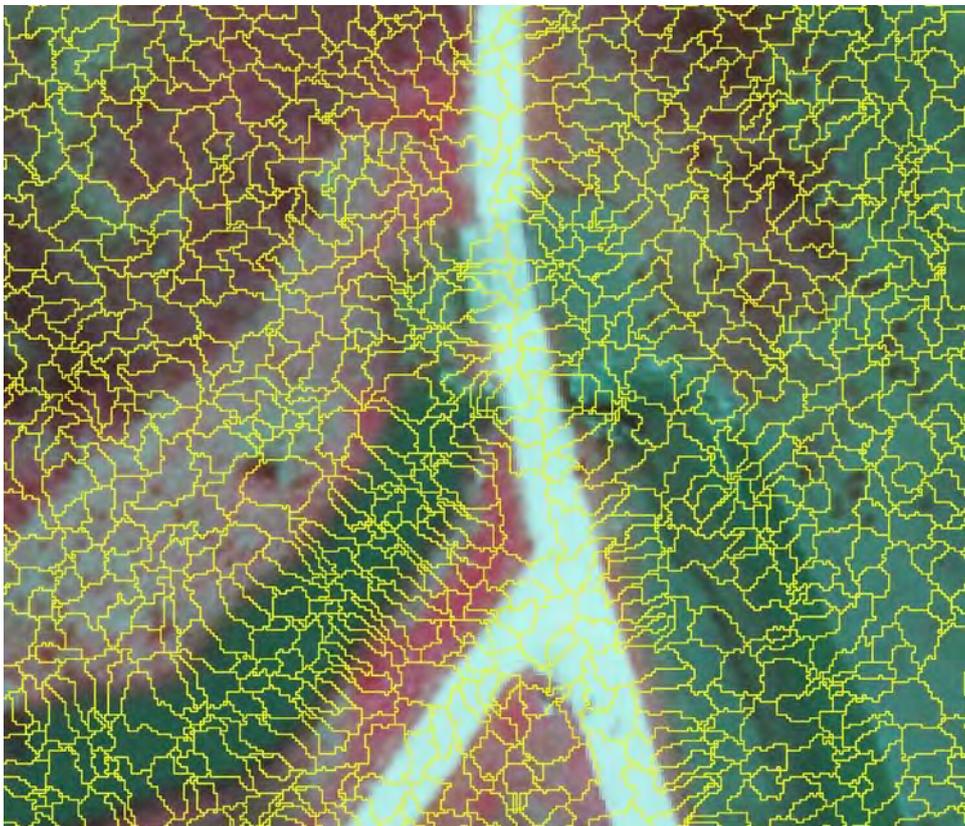
⁸ <https://www.epfl.ch/labs/ivrl/research/slic-superpixels/>

versions de SLIC puisque l'on ne peut pas exporter les résultats dans la version d'essai.

SLIC est également disponible dans un addon de GRASS⁹ où le code original est traduit en C, et dans un module d'Orfeo Toolbox¹⁰.

L'algorithme Watershed est disponible également dans SAGA et GRASS. Il vise à délimiter des bassins versants et nécessite pour fonctionner correctement un MNT en entrée.

Concernant l'algorithme **Watershed de GRASS**, en l'absence de MNT, les contours produits ne sont pas concluants. Les contours vont être situés au milieu de zone relativement homogènes et non aux bordures. Son utilisation nécessite donc davantage de données en entrée, ce que l'on ne souhaite pas forcément. La [documentation](#) est assez fournie, bien que certains [tutoriels](#) soient plus indiqués pour fournir des exemples d'utilisation.

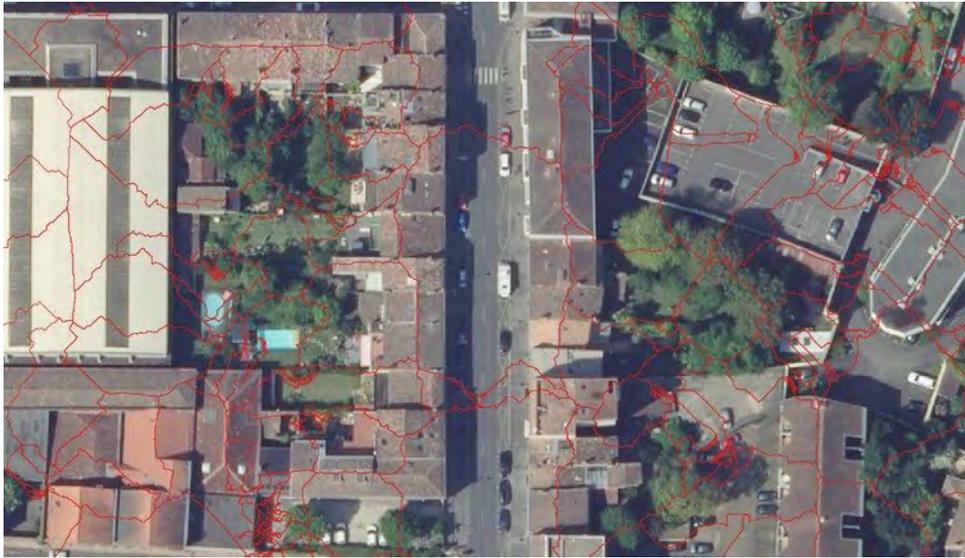


Watershed GRASS sur image rvb (©IGN)

En utilisant une image 5 canaux : R, V, B, IR et MNS-MNT, le résultat n'est pas plus concluant, ce qui est logique puisque l'information de hauteur est bien différente de celle attendue. Or, c'est ce type de donnée qui est préparée et utilisée. L'algorithme de GRASS nécessiterait donc d'utiliser une autre source de données, ce qui n'est pas envisagé pour le moment car cela densifierait les pré-traitements.

9 <https://trac.osgeo.org/grass/browser/grass-addons/grass7/imagery/i.superpixels.slic>

10 https://gitlab.orfeo-toolbox.org/remote_modules/otb-slic



Watershed GRASS sur la couche MNH (©IGN)

Le souci est le même dans SAGA, le Watershed est difficile à manier pour obtenir des résultats satisfaisants sans MNT, la définition des contours ne correspond pas au besoin.

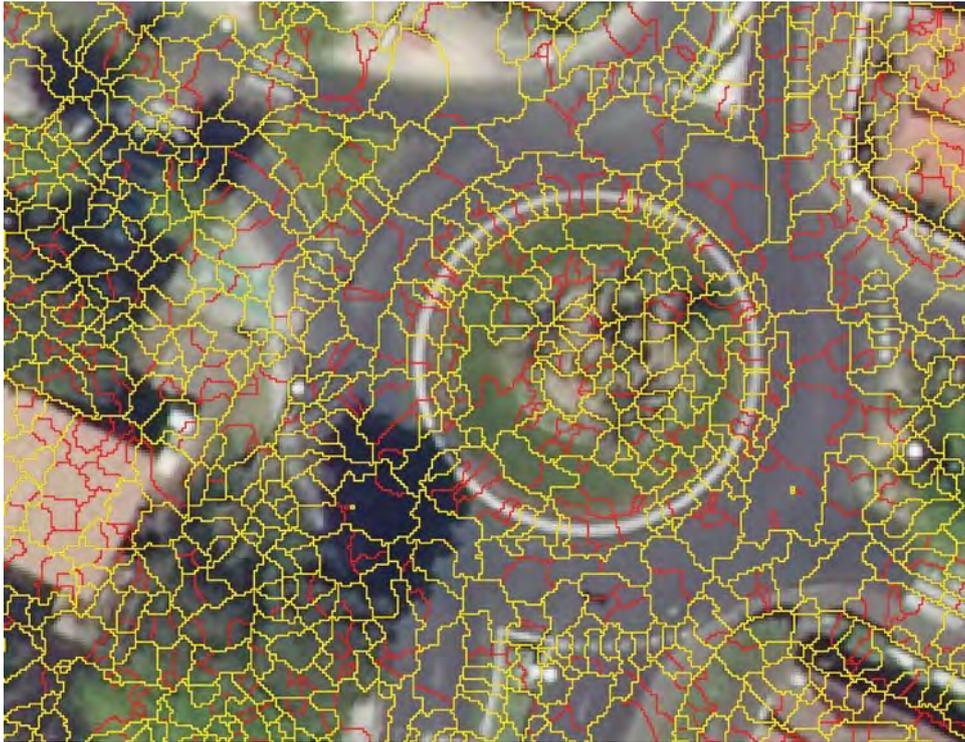


SAGA algorithme watershed appliqué en prenant en compte les maxima et ayant pour valeur de seuil 5. En jaune, la segmentation effectuée sur une image IRC et en noir sur une image RVB. (©IGN)

Les bords des objets (bâtiments, routes) ne sont pas très nets. Les bordures sont souvent au milieu d'une zone plus sombre ou plus claire et non au bord de ces zones.

Par contre, une hiérarchie de segmentation peut être réalisée en gardant les mêmes paramètres et en changeant uniquement la valeur de seuil pour regrouper des pixels afin de former des régions. Les plus petits segments sont

inclus et partagent des bords avec les segmentations plus grossières réalisées indépendamment.



Watershed de SAGA avec un seuil de 5 en jaune et de 2 en rouge (©IGN)

Apport annotation : les algorithmes Watershed de GRASS et SAGA ne sont pas utilisables en l'état sur les données en entrée.

Paramétrage sur différents types de sol

Le paramétrage étant une étape longue à réaliser mais déterminante pour les résultats et la précision de la classification en sortie, il convient de déterminer si les algorithmes rencontrés sont faciles à paramétrer, à implémenter dans les outils IGN et s'ils sont de qualité suffisante par rapport aux attentes.

Les temps de calculs mesurés par algorithme sont réalisés sur une orthophotographie rvb de 300 * 200 mètres (4,5 MB). La machine de calcul utilisée a les paramètres suivants : système linux, processeur AMD® Ryzen 5 4500u, radeon graphics x 6 et mémoire 7,2GiB.

Les jeux tests sont disponibles dans un dossier « data » joint à un jupyter-notebook présentant les algorithmes de la librairie Scikit-image. Ces algorithmes sont également accessibles dans QGIS à partir du plugin [Terminus](#) et dans la bibliothèque logicielle de télédétection et de SIG (RSGISlib).

Le nombre de paramètres des algorithmes non présentés dans cette étude mais permettant de modifier le résultat en sortie sont indiqués entre parenthèses. Les algorithmes sont détaillés dans l'annexe *Description des algorithmes*. Felzenszwalb, Quickshift, SLIC et le Watershed de la bibliothèque scikit-image appartiennent à la famille des superpixels.

	Entrée	Temps de calcul	Paramètres	Complexité	Documentation
Felzenszwalb	Multi-canaux (3 et +)	4.18 s ± 105 ms	3	$O(n \log(n))$	Doc
Quickshift	3 canaux (RVB)	6.68 s ± 27.4 ms	3 (+3)	$O(dn^2)$ avec d la dimension de n	Doc
SLIC	Multi-canaux (3 et +)	1.45 s ± 11.3 ms	3 (+7)	$O(n)$	Doc
Watershed	2 et +	0.72 s ± 19.5 ms	2 (+3)		Doc
Watershed - GRASS	MNT	3.72 s algo + 0.21 s gdal_polygonize	2 et +		Code Doc
Watershed - SAGA	MNT				Doc
LargeScaleMean Shift	1 et +	13.02 s	3 et +		Doc

Tableau des caractéristiques des algorithmes de segmentations

Les algorithmes de segmentations ont été testés sur différents types de sol : des cultures labourées, des plantations d'arbres, des bâtiments afin d'observer brièvement si certains algorithmes sont plus pertinents pour segmenter un certain type d'occupation du sol. Bien sûr, c'est une étude qui mériterait d'être approfondie en soi. Il s'agit ici d'illustrer surtout les différents comportements des algorithmes selon l'occupation du sol.

Des comparaisons ont été opérées entre les algorithmes Felzenszwalb, Quickshift, SLIC et Watershed de la bibliothèque Scikit-image. Les paramètres définis pour les algorithmes sont les mêmes appliqués à différentes images.

Le résultat le plus différent entre les algorithmes s'observe pour des cultures, notamment des sols labourés. Les stries des champs influent beaucoup plus le découpage opéré par un algorithme Watershed ou Felzenszwalb contrairement à un algorithme SLIC où l'on retrouve bien le maillage initial.

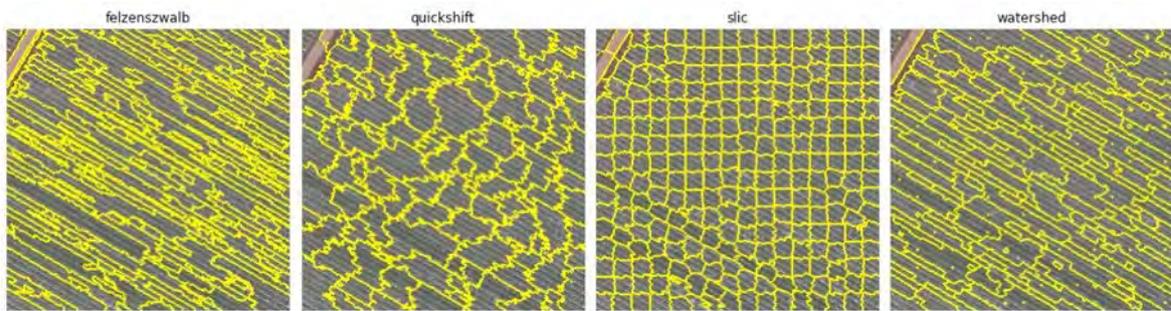


Illustration résultats segmentations sur champs (©IGN)

Au contraire, pour délimiter les contours d'arbres plantés régulièrement, l'algorithme fournit des résultats assez similaires.

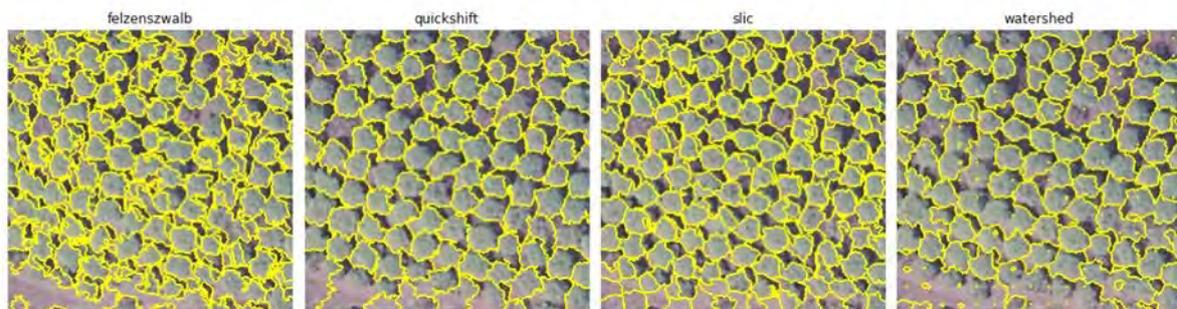


Illustration résultats segmentations sur arbres (©IGN)

Pour délimiter les bâtiments, l'algorithme Felzenszwalb se distingue des trois autres en délimitant des zones de couleurs moins homogènes que les autres algorithmes. SLIC montre également des imprécisions dans la délimitation de zones homogènes.

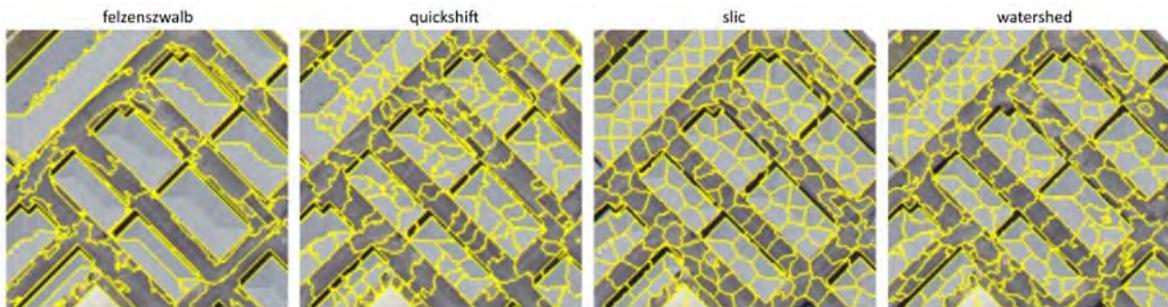


Illustration résultats segmentations sur bâtiments (©IGN)

Ce n'est qu'une brève ébauche des différences observables entre algorithmes selon l'occupation du sol. Elle vise à illustrer la difficulté de réglage d'un paramètre pour qu'il capture tous les éléments d'une zone. On comprend également qu'il est difficile de définir une configuration de segmentation qui convienne à l'ensemble d'une zone ou d'un territoire étant donné la grande variabilité d'objets s'y trouvant.

La grande variabilité de chaque classe (d'homogénéité, de texture, de taille, d'heure de prise de vue aérienne des différentes images, de contraste, d'aspect) fait qu'il est impossible de trouver un paramétrage optimal pour l'ensemble de la zone.

Apport annotation : facilité d'implémentation des algorithmes de la bibliothèque Scikit-image, comparaison facile, documentation fournie, peu de paramètres à manipuler.

Selon Andrés Troya-Galvis du laboratoire Icube, « Il n'existe pas d'algorithme (ou paramétrage) permettant de segmenter correctement la totalité des objets d'intérêt. Il semble prometteur de chercher une segmentation spécialisée à chaque classe thématique. ». A terme, il pourrait être intéressant de suivre les pistes de ce laboratoire qui recommande d'apprendre des seuils adaptés à chaque classe thématique, et de définir une stratégie de collaboration segmentation/classification. Cette dernière doit pouvoir permettre d'automatiser davantage le processus d'annotation de la donnée.

Sensibilité de l'algorithme

Stabilité

L'un des problèmes rencontrés dans cette étude est le **défaut d'invariance** de certains algorithmes. Or, il paraît nécessaire qu'un algorithme soit stable pour être fiable et pouvoir le paramétrer correctement. Voici deux définitions plus intuitives que mathématiques de la stabilité d'un algorithme :

« Étant donné une image, étant donné un objet d'intérêt dans cette image, un algorithme de segmentation est considéré comme stable si tout sous-ensemble d'image contenant cet objet engendre exactement le même résultat de segmentation de cet objet ». « Étant donné une image, étant donné deux sous-ensembles se chevauchant de cette image, un algorithme de segmentation est considéré comme stable si les résultats de segmentation à l'intersection de ces deux sous-ensembles d'image correspondent exactement. » ¹¹

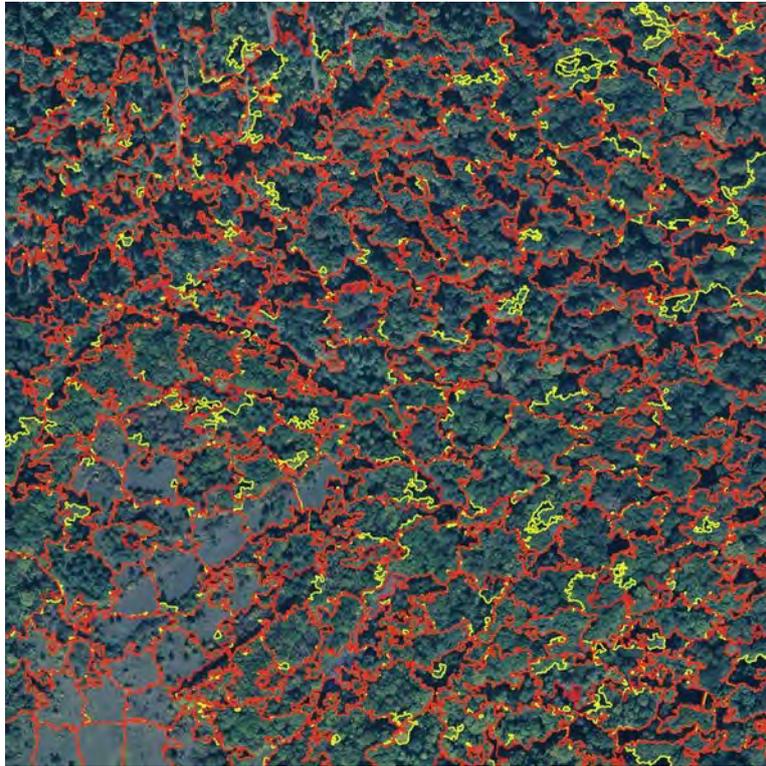
Les algorithmes présentés dans cette étude valident cette définition de la stabilité.

Par ailleurs, on va regarder si les algorithmes sont stables sur des images de même emprise mais renvoyant des informations différentes (variations radiométriques ou compression de l'image).

Compression d'une image

En premier test, on applique un algorithme avec certains paramètres à une image rvb ainsi qu'à la version compressée de cette image. On obtient une segmentation très similaire qui n'a que de légères différences dans le choix des contours de régions. Les changements n'étant pas diamétralement différents, l'utilisation d'images compressées ne changera pas significativement le résultat de la segmentation. C'est intéressant pour gagner en temps de calcul sur de grandes zones d'étude. Sur de petites zones, les temps de calcul sont du même ordre de grandeur.

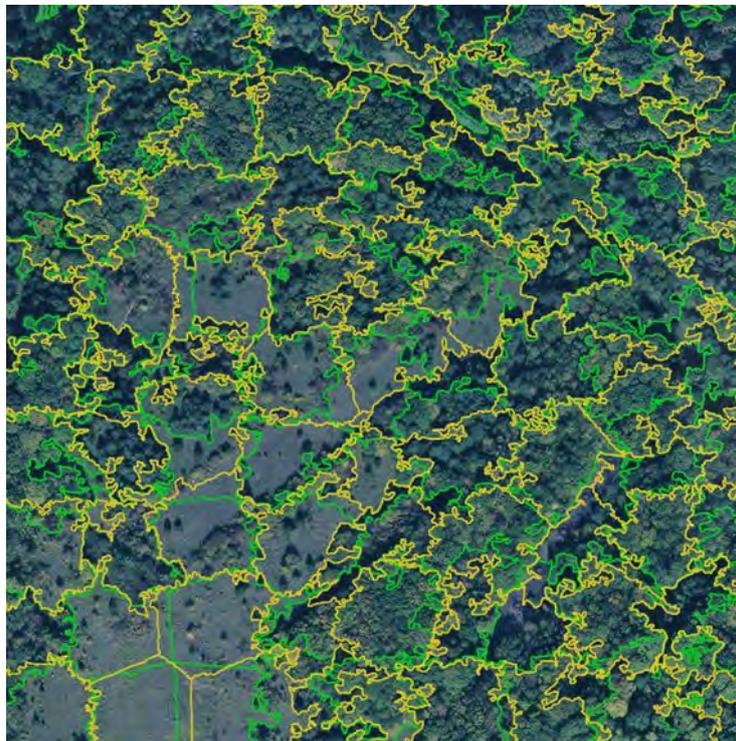
11 Julien Michel, David Youssefi, and Manuel Grizonnet, « Stable Mean-Shift Algorithm and Its Application to the Segmentation of Arbitrarily Large Remote Sensing Images ». IEEE, 2015.



Segmentations SLIC avec des paramètres identiques sur image compressée en rouge et sur image non compressée en jaune ©IGN

Images multispectrales

Le deuxième test consiste à appliquer un algorithme avec certains paramètres à une image rvb ainsi qu'à une image irc de la même emprise. On obtient des segmentations assez différentes.



Segmentations SLIC avec des paramètres identiques sur une image rvb en jaune et sur une image irc en vert ©IGN

Certains algorithmes de la librairie Scikit-image ont un paramètre pour prendre en compte ou non l'aspect multi-bande des orthophotographies.

Algorithme	Paramètre multichannel=True	Non utilisation du paramètre
Felzenszwalb	24.7 s ± 367 ms	31.2 s ± 1.61 s
SLIC	24.1 s ± 1.03 s	49.5 s ± 7.56 s

Dans les deux cas il est intéressant d'utiliser ce paramètre pour gagner en temps de calcul, sachant que par ailleurs, les résultats de segmentation obtenus sont strictement les mêmes.

Apport annotation : une segmentation appliquée à une image compressée renvoie un résultat légèrement différent de l'image originale mais qui n'est pas aberrant pour les besoins d'annotation.

Utilisation sur de grandes images

Dans le processus d'annotation des données d'apprentissage, il n'est pas forcément nécessaire d'avoir des algorithmes adaptés à de grandes images puisque l'on travaille généralement sur des tuiles ou des vignettes. Pour autant, cela peut intéresser un annotateur s'il veut travailler sur une grande zone sans la découper pour garder du contexte.

Cette problématique est abordée dans le cadre d'une segmentation par les algorithmes LargeScaleMeanShift d'Orfeo Toolbox, ainsi que par celui de Shepherd et al.

L'algorithme LargeScaleMeanShift est très cité dans l'état de l'art¹². Il permet de tuiler l'image en entrée et de gérer la fusion des tuiles une fois les segmentations réalisées en fin de traitement. Il peut donc s'appliquer sur de grandes images, attention tout de fois aux temps de calcul qui peuvent être gourmands.

L'algorithme de [Shepherd et al., 2019] est disponible sous RSGISlib. Le gros avantage de cet algorithme est qu'il fonctionne bien sur de grandes images sans création au préalable de mosaïques, avec comme exemple la région de Canterbury en Nouvelle-Zélande qui fait plus de 44000 km². Il y a donc une

¹² Chaabouni-Chouayakh, H.; Datcu, M. Coarse-to-fine approach for urban area interpretation using TerraSAR-X data. IEEE Geosci. Remote Sens. Soc. Newsl. 2010, 7, 78-82.

Chehata, N.; Orny, C.; Boukir, S.; Guyon, D.; Wigneron, J.P. Object-based change detection in wind storm-damaged forest using high-resolution multispectral images. Int. J. Remote Sens. 2014, 35, 4758-4777.

Bellakanji, A.C.; Zribi, M.; Lili-Chabaane, Z.; Mougnot, B. Forecasting of Cereal Yields in a Semi-arid Area Using the Simple Algorithm for Yield Estimation (SAFY) Agro-Meteorological Model Combined with Optical SPOT/HRV Images. Sensors 2018, 18, 2138.

économie de temps de calcul et d'opérations complexes telle la fusion des limites des tuiles en fin de traitement. Ils ont pris pour ce faire des images SPOT-5 et Sentinel-2 avec pour paramètres 60 clusters en entrée et une unité cartographique minimale de 100 pixels. Pour un volume de 60M pixels (et 4 bandes spectrales) les calculs ont pris 3 heures sur un processeur x86 de 3GHZ avec 12 Go de RAM sous Linux, résultant au total à 1,2M de segments/polygones.

Une autre version de l'algorithme a été proposée afin de tuiler les images Landsat et AOS PALSAR en entrée et de gérer la fusion des tuiles en fin de traitement [Shepherd et al., 2019].

Apport annotation : les algorithmes [Shepherd et al.] et LargeScaleMeanShift (Orfeo Toolbox) sont utiles si l'on souhaite l'appliquer directement à une grande image et s'épargner ainsi des traitements supplémentaires de découpe en tuiles et fusion de ces tuiles.

Segmentation multi-échelle

Le plus gros problème rencontré en segmentation est le **paramétrage** : il est difficile d'en déterminer un qui permette de segmenter correctement l'ensemble d'une scène. En effet, à une certaine échelle d'analyse, si l'on prend l'exemple des cultures, certaines limites peuvent ne pas être respectées : certains champs homogènes et voisins vont être regroupés en une seule région, tandis que d'autres parcelles très texturées peuvent être découpées en plusieurs régions. Elles peuvent également être de tailles très diverses, ce à quoi ne s'adaptent pas tous les algorithmes. Cette variabilité propre à chaque classe de la nomenclature de l'OCSGE tend à modifier l'approche de la segmentation utilisée.

Plutôt que de rechercher un paramétrage concluant à une certaine échelle d'analyse, il apparaît plus pertinent de répondre à la question : « à quelle échelle trouve-t-on des formes qui nous intéressent ? » [Guigues et al., 2003]. Segmenter une image étant un **problème d'échelle**, il apparaît pertinent de définir une **structure multi-échelle** de l'image pour capturer les éléments qui nous intéressent à différents niveaux. L'annotation de données d'apprentissage se fait alors en deux temps :

- La création d'une hiérarchie de segmentation sur la zone à annoter
- L'étiquetage par un photo-interprète qui choisit l'échelle la plus pertinente pour saisir les différents éléments : reconnaître et saisir la géométrie des objets

Cette partie présente 3 cas de réalisation d'une segmentation multi-échelle :

- L'utilisation d'un algorithme qui génère une pyramide de segmentation
- L'utilisation d'un algorithme stable et utilisé en ne modifiant qu'un seul de ses paramètres pour générer une hiérarchie de segmentations
- La réalisation d'une fusion de segments

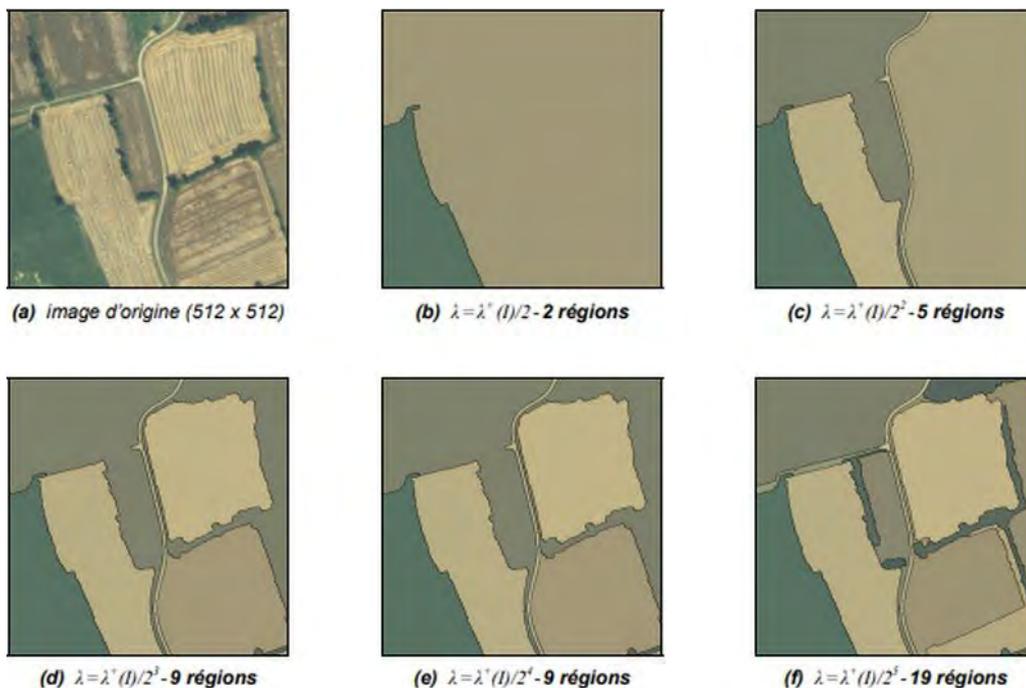
Algorithme hiérarchique

Les algorithmes de segmentation hiérarchiques fonctionnent principalement par fusion itératives de segments. Les segments initiaux sont les pixels de l'image ou des groupes de pixels, les algorithmes varient alors sur le critère du choix dans l'ordre des segments à fusionner. Cela peut être fait en définissant une énergie (score) pour chaque segment et en cherchant à fusionner en priorité les deux segments qui entraînent la plus faible augmentation de l'énergie totale. La définition de l'énergie d'un segment peut dépendre alors de plusieurs critères dont sa forme (compacité) et son accroche aux données (distance de la valeur du segment aux valeurs des pixels associés sur l'image). Une segmentation donnée est alors définie par un niveau d'énergie et correspond à une coupe dans l'arbre de fusion de région.



Niveaux hiérarchiques de la segmentation Pyram (© L Guigue , [Scale set image analysis](#))

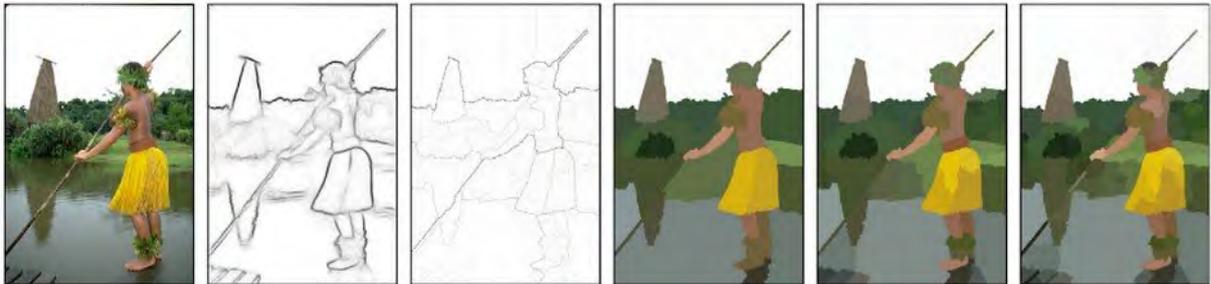
L'algorithme multi-échelle de [Guigues et al., 2003] est un algorithme de segmentation hiérarchique utilisé au sein de l'IGN et implémenté dans l'outil nommé Pyram. Il a été testé tout d'abord sur des orthophotographies sous-échantillonnées à 1 mètre d'un département français (taille de 100 000 par 100 000 pixels environ), ce qui a pris 2 jours de calculs avec une parallélisation sur 12 Macintosh G4 bi-processeurs à 1 Ghz.



Résultat de segmentation avec l'algorithme pyram selon différent niveau d'énergie (coupe dans la hiérarchie)(© L Guigue , [Scale set image analysis](#))

Cet algorithme fournit plusieurs niveaux de segmentation selon les spécifications de l'utilisateur. Il assure une cohérence topologique et géométrique des données.

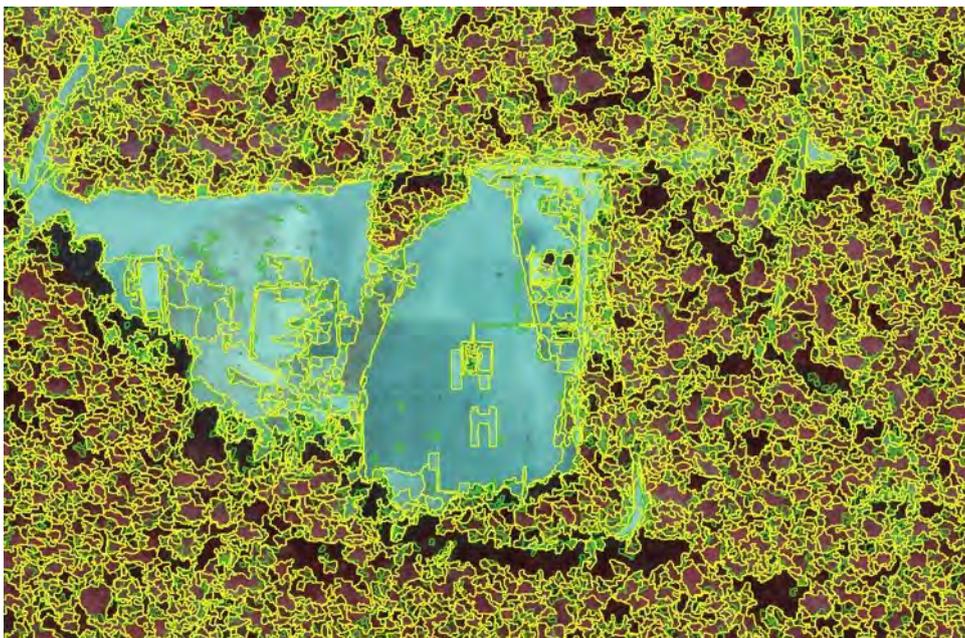
Ces algorithmes de segmentations hiérarchiques reviennent à des problématiques de construction d'arbres et minimisation sur des graphes. On peut donc utiliser des bibliothèques dédiées pour cela afin de calculer ces segmentations. Un exemple d'une telle bibliothèque est [Higra](#) ("Hierarchical Graph Analysis") qui est open-source et utilisable en python. Elle possède des exemples d'utilisation dans le cadre de la segmentation d'image et permet d'implémenter facilement le même algorithme que `pyram` (voir fonction [labelisation_optimal_cut_from_energy](#))



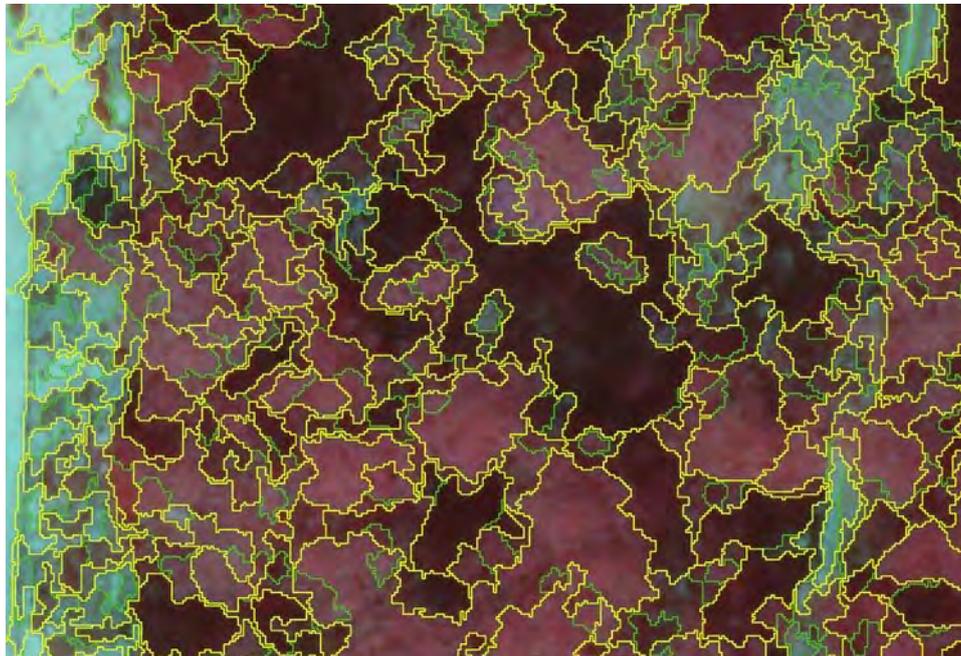
Segmentation d'image avec Higra. De gauche à droite, image originale, gradient, carte de niveau/saillance dans la hiérarchie calculée, image reconstruite (aplat de couleur) avec 25, 50 et 100 segments. (© B Perret, [Higra Hierarchical graph Analysis](#))

Algorithme stable utilisé pour produire une hiérarchie de segmentation LargeScaleMeanShift d'Orfeo Toolbox

Cet algorithme peut être utilisé pour réaliser une segmentation hiérarchique à condition d'utiliser toujours le même paramétrage et de ne modifier qu'un seul d'entre eux. Ainsi, une fois les paramètres *spatial* et *ranger* définis, si l'on ne modifie que la taille minimale des segments, on peut obtenir des segmentations qui s'emboîtent les unes dans les autres (comportement observé mais pas encore étudié pour confirmation théorique).



Algorithme *LargeScaleMeanShift* d'Orfeo Toolbox sur une zone forestière et sol nu (© IGN)



Segmentations LargeScaleMeanShift d'Orfeo Toolbox et zoom sur un détail. Paramètres identiques, hormis la taille minimale des segments : valeur de 50 pour la segmentation en vert et de 200 pour la segmentation en jaune (© IGN)

Multi-résolution d'eCognition

Cet algorithme est très utilisé dans l'état de l'art¹³. Il repose sur des critères d'hétérogénéité détaillés en annexe A – Description des algorithmes. Il semble efficace pour générer une hiérarchie de segments.

Il est présenté pour avoir une étendue de l'existant mais n'est pas exploré plus que cela, car il nécessite une licence d'utilisation. Une [étude](#) a comparé les paramètres disponibles de l'algorithme multi-résolution d'eCognition et réalisé une hiérarchie de segmentations.

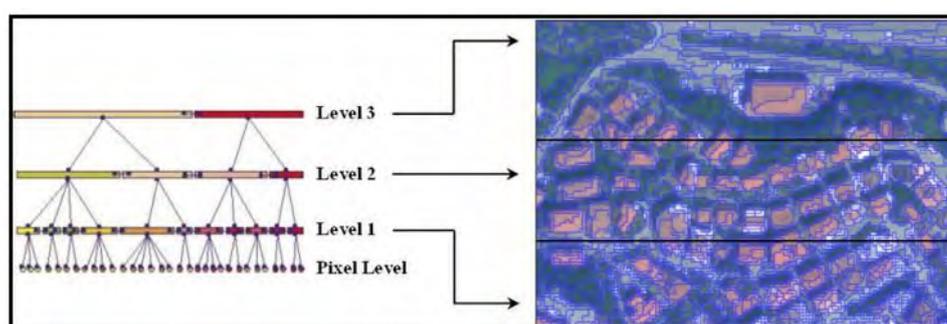
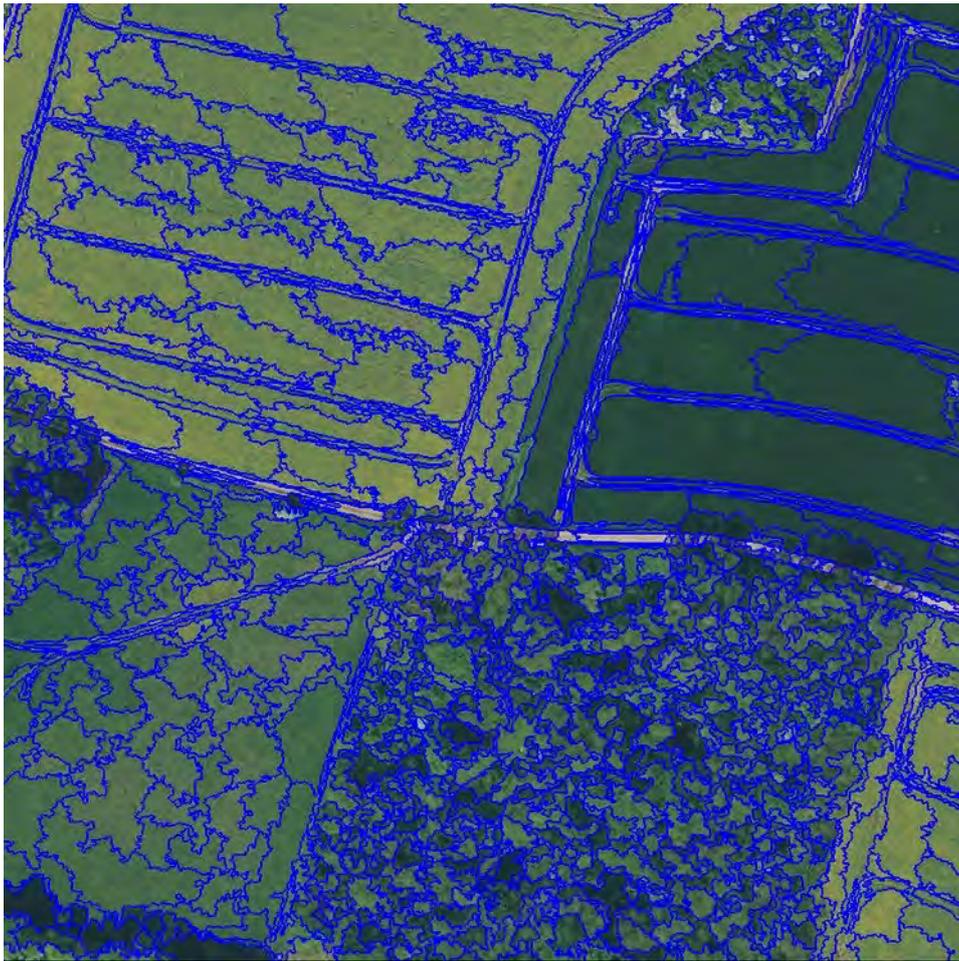


Fig. 3: Hierarchical net of image objects derived from image segmentation level 1 (5 pixels scale parameter), level 2 (10 pixels scale parameter) and level 3 (20 pixels scale parameter) in Project 2.

(© S Karakis , [Analysis of segmentation parameters in ecognition software](#))

13 Lucas, R.; Rowlands, A.; Brown, A.; Keyworth, S.; Bunting, P. Rule-based classification of multi-temporal satellite imagery for habitat and agricultural land cover mapping. ISPRS J. Photogramm. Remote Sens. 2007, 62, 165–185.



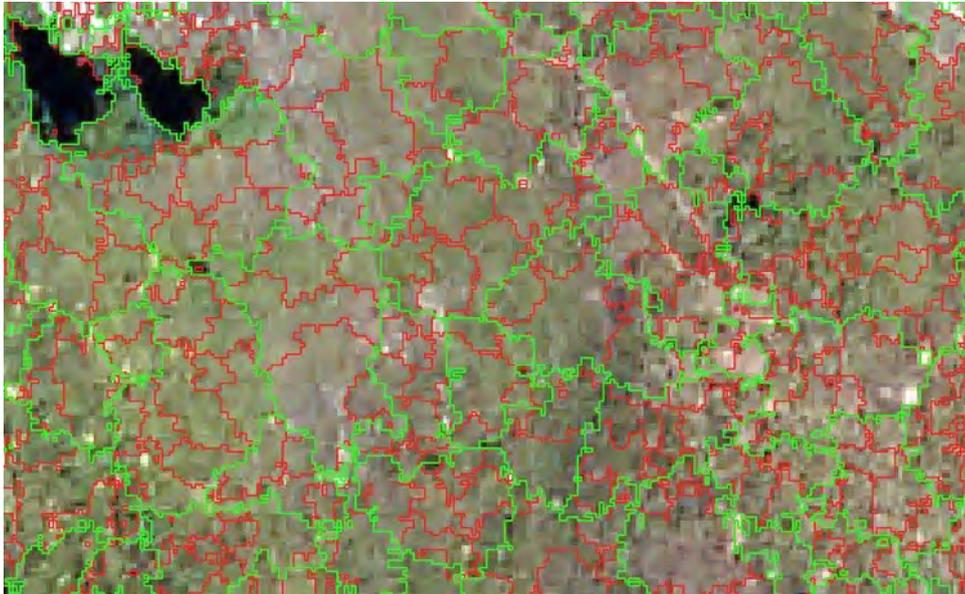
Algorithme multi-résolution ECognition

Construction d'une hiérarchie

Fusion de segmentations RAG Scikit-image

La librairie Scikit-image propose un algorithme de jointure de segmentations. Il considère que deux pixels sont dans le même segment si et seulement s'ils sont dans le même segment dans S1 et S2. Il a été implémenté pour évaluer sa pertinence face à des algorithmes de segmentation hiérarchique.

Étant donné que l'algorithme effectue une simple fusion de segmentations existantes, il ne permet pas d'obtenir des résultats aussi bons que les segmentations de type hiérarchique à plusieurs niveaux. On ne peut pas réaliser de segmentations imbriquées les unes dans les autres avec cet algorithme.



Simulation d'une segmentation de niveau 1 en vert et d'une segmentation de niveau deux en rouge. Algorithme Quickshift en zone naturelle et forestière (© IGN)

On constate que les deux segmentations ont certains contours en commun mais que les géométries sont bien trop différentes pour que la superposition soit propre et que cela puisse être utilisé dans le cadre d'une segmentation hiérarchique.

L'écart est flagrant avec d'autres algorithmes comme SLIC, où l'on constate que la modification des paramètres entraîne des changements bien trop importants dans le contour des régions du fait du fonctionnement intrinsèque de l'algorithme. Celui-ci commence par créer une grille de centres de clusters en fonction du nombre de régions souhaitées. On comprend bien que réaliser des segmentations imbriquées avec un simple algorithme de fusion n'est pas possible avec SLIC, ce qui vaut pour Quickshift, Felzenszwalb et Watershed.



Superposition de deux segmentations SLIC où l'on a simplement augmenté le nombre de régions souhaitées pour la segmentation en rouge (© IGN)

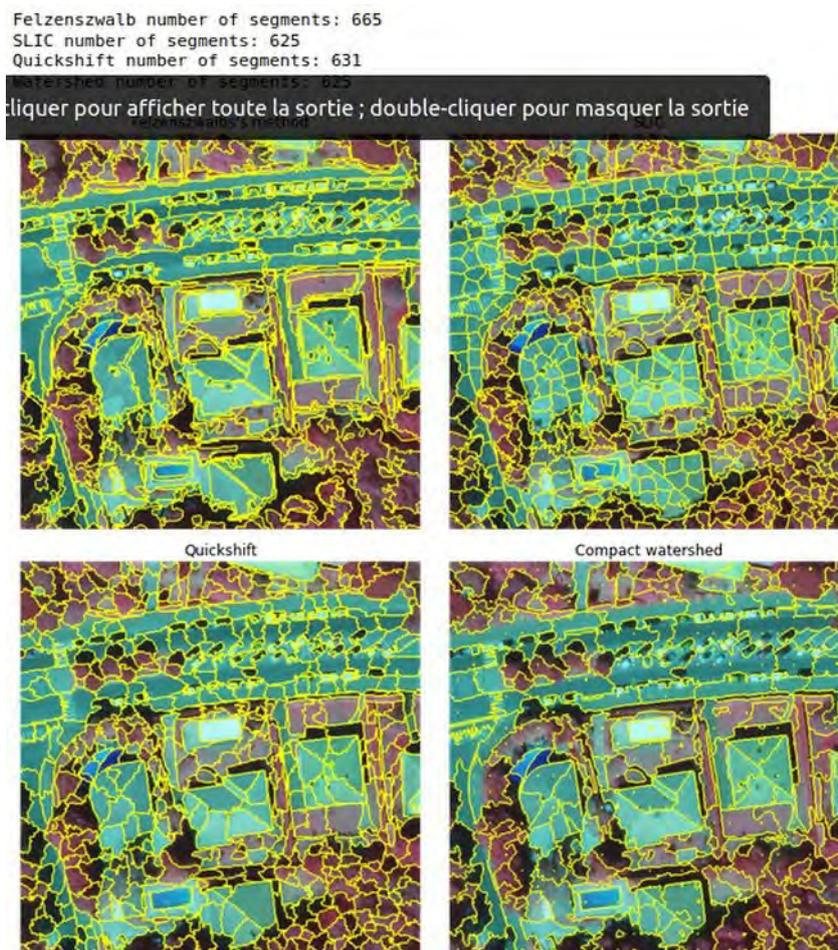
Apport annotation : cette création de hiérarchie par fusion de segmentation superpixel/SLIC est la moins concluante

Conclusion : Face aux difficultés de paramétrage d'un algorithme, il apparaît préférable de déployer un algorithme de segmentation multi-échelle et de laisser plus de **marges de manoeuvre à l'annotateur** dans le choix des niveaux de visualisation et de capture de l'information pertinente en fonction des attentes. Il convient alors d'avoir un outil d'annotation suffisamment performant pour réaliser cette deuxième phase, afin que l'exploration des différents niveaux de segmentation soit le plus fluide possible. Il y a un gros intérêt de visualisation dynamique, et de zoom et dé-zoom dans l'image à travers les niveaux de la hiérarchie.

Comparaison des algorithmes rencontrés

Il existe déjà des benchmarks de comparaison d'algorithmes de segmentation mais qui s'appliquent surtout à des images de computer vision. Cette étude vise davantage à cerner les apports de ces algorithmes appliqués à des images aériennes ou satellitaires.

Il est facile de comparer les algorithmes de la bibliothèque scikit-image à travers le jupyter-notebook fourni comme présenté pour la segmentation de différents types de sols. Ci-dessous, les quatre algorithmes sur des bâtiments dans un quartier résidentiel.



Algorithmes de la bibliothèque Scikit-image

Un article intéressant compare l'algorithme de Shepherd et al. Implémenté dans la librairie RGISlib avec les algorithmes croisés dans cette étude : le meanshift implémenté dans Orfeo Toolbox, le multi-résolution d'eCognition ainsi que les algorithmes Quickshift et Felzenszwalb implémentés dans la librairie Scikit-image sur une scène SPOT-5 de 10m de résolution spatiale.

Les auteurs James D. Shepherd, Pete Bunting et John R. Dymond ont testé de nombreux paramètres pour chaque segmentation.

Table 3. Parameters used for each of the segmentation algorithms.

Algorithm	Parameters	Number of Segmentations
eCognition	scale: [10–100], shape: [0–1], compact.: [0–1]	1210
Mean-Shift	range radius; [5–25], convergence thres.: [0.01–0.5], max. iter.: [10–500], min. size: [10–500]	625
Felzenszwalb	scale: [0.25–10], sigma: [0.2–1.4], min. size: [5–500]	343
Quickshift	ratio: [0.1–1], kernel size: [1–20], max. dist.: [1–30], sigma: [0–5]	1500
Shepherd et al.	k: [5–120], d: [10–10,000], min. size: [5–200]	540

(© D. Shepherd, Pete Bunting: [Operational Large-Scale Segmentation of Imagery Based on Iterative Elimination](#))

Il en ressort que l'algorithme de Shepherd obtient les meilleurs résultats en qualité. Il montre de bonnes performances en termes d'utilisation de la mémoire et de temps de calcul. Accessible dans la bibliothèque RSGISIS, les auteurs le recommandent pour segmenter les images satellitaires optique et radar.

Table 4. Comparison of the segmentation results.

Algorithm	Parameters	Rank	gs_f	f	Precision	Recall	gs	gs ^{green}	gs ^{red}	gs ^{nir}	gs ^{swir}
Shepherd et al.	k: 60, d: 10,000, min. size: 10	1	0.74	0.85	0.84	0.86	0.97	1.01	0.90	0.99	0.98
Quickshift	ratio: 0.75, kernel size: 10, max. dist.: 5, sigma: 0, lab colour space.	86	0.64	0.73	0.80	0.68	0.94	0.92	0.86	0.99	0.98
Mean-Shift	range radius; 15, convergence thres.: 0.2, max. iter.: 100, min. size: 10	253	0.56	0.61	0.55	0.70	0.93	0.95	0.87	0.96	0.94
eCognition	scale: 10, shape: 0.7, compact: 0.2	411	0.49	0.52	0.57	0.48	0.95	0.99	0.92	0.95	0.93
Felzenszwalb	scale: 10, sigma: 12, min. size: 20	539	0.47	0.46	0.49	0.43	1.10	1.14	1.04	1.17	1.04

(© D. Shepherd, Pete Bunting: [Operational Large-Scale Segmentation of Imagery Based on Iterative Elimination](#))

L'algorithme de Shepherd et al. est le mieux classé en utilisant une métrique, gs_f, qui moyenne la qualité extrinsèque (comparaison à une segmentation manuelle) et intrinsèque (variance radiométrie intra et inter-segment) de la segmentation. Le terme Rank donne une idée du classement des segmentations utilisant différents paramétrages les uns par rapport aux autres. Ainsi, 252 segmentations de Shepherd et al. et de Quickshift ont eu de meilleurs résultats que la meilleure segmentation Mean-Shift.

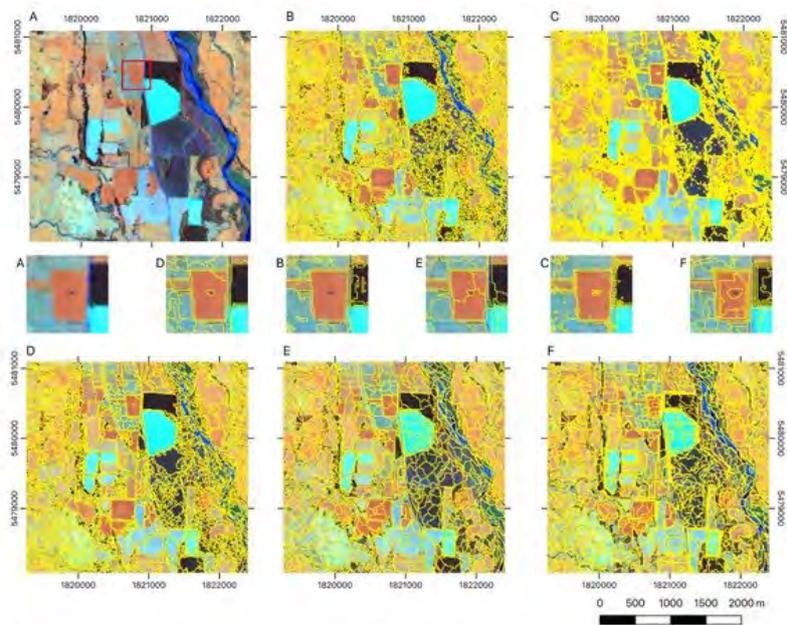


Figure 10. Comparison of segmentation algorithms with the parameters found in Table 4; the red box in (A) highlights the region of the zoom-in area. (A) SPOT-5 image with NIR, SWIR, red band combination, (B) Shepherd et al. (this article), (C) Quickshift [52], (D) mean-shift [42], (E) eCognition [28] and (F) Felzenszwalb [53].

figure 10 article D. Shepherd, Pete Bunting: [Operational Large-Scale Segmentation of Imagery Based on Iterative Elimination](#) ©

Selon la figure 10 de l'article, les algorithmes de Shepherd et al. et de Mean-Shift semblent plus appropriés visuellement et ont une qualité de segmentation similaire. On peut y voir que la plupart des zones homogènes sont bien détournées en segments continus et correspondent aux régions de référence. Les zones de variation spectrale sont bien délimitées.

La segmentation multi-résolution d'eCognition est assez satisfaisante mais la propriété de l'algorithme veillant à produire des segments de taille similaire induit une forte sur-segmentation des zones homogènes par rapport aux algorithmes de Shepherd et al. et de Mean-Shift. On constate par ailleurs, que dans les régions à très forte variation spectrale comme les bords de bâtiments, l'algorithme Quickshift produit de nombreux petits segments tandis que l'algorithme Felzenszwalb produit des segments très longs et fins.

D'autres études ont comparé des algorithmes entre eux. Ainsi, dans la [documentation](#) de l'algorithme Watershed implémenté par GRASS, on apprend que les auteurs ont comparé leur algorithme avec le Watershed d'ArcGIS. Ils en concluent une meilleure performance de la segmentation de GRASS pour délimiter les cours d'eau et les demi-bassins versants, pour segmenter les pentes plates et faibles, et pour ne pas confondre les sommets de la canopée avec l'altitude du sol sur des zones où le couvert forestier est irrégulier et ne couvre pas toute la zone.

Pour aller plus loin dans l'analyse, il resterait surtout à comparer l'algorithme de Shepherd et al. avec l'outil IGN Pyram.

Avantage général des superpixels : ils réduisent fortement le nombre de primitives à partir desquelles calculer les caractéristiques de l'image, ce qui réduit la complexité des traitements ultérieurs.

Algorithme	Avantages	Inconvénients
Felzenszwalb	Préserve les détails dans les régions de l'image à faible variabilité et l'ignore dans les régions à forte variabilité. Suffisamment rapide pour pouvoir être utilisé sur des images à grande échelle (Ratan et al. 1999)	Pas moyen de contrôler le nombre ou la compacité des superpixels directement (R. Achanta et al., 2010)
Quickshift - Scikit-image	L'algorithme est simple, rapide et permet d'équilibrer entre sous et sur-segmentation des modes. Plus rapide que le Mean-shift	Pas moyen de contrôler le nombre ou la taille des superpixels directement (R. Achanta et al., 2010).
SLIC	Efficacité de calcul : produit des segments compactes et presque uniformes à un faible coût. Visuel significatif : similarité couleurs ou textures dans un superpixel. Sur-segmentation : très peu de perte de pixels lors passage grille de pixels aux superpixels. Il permet d'obtenir une régularité dans les formes des superpixels et une bonne compacité.	Faible qualité de la segmentation. L'utilisateur définit le paramètre de compacité qui est donc le même pour l'ensemble de l'image. SLIC a donc tendance à produire des superpixels de taille régulière dans les régions lisses et d'autres de taille très irrégulière dans les régions texturées. Les aspects de l'image ne sont pas assez pris en compte
Watershed - Scikit-image		Le bruit ou des irrégularités locales entraînent des sur-segmentations. Pour éviter ce problème, soit un pré-traitement du bruit de l'image est réalisé, soit les régions sont fusionnées après la segmentation selon un critère de similarité
Watershed - GRASS	Précis dans les zones de faible pentes, capable de reproduire l'emplacement des cours d'eau dans les grandes vallées, même si le couvert forestier est important.	
Meanshift	Pas besoin de définir le nombre de segments à l'avance. Il peut également s'appliquer sur un	Pas moyen de contrôler le nombre ou la taille des superpixels directement (R.

	espace de points quelle qu'en soit la dimension, ce qui est intéressant si l'on veut prendre en compte plusieurs bandes d'un raster avec des indices spectraux	Achanta et al., 2010)
LargeScaleMeanShift - Orfeo Toolbox	Ceux du Meanshift. De plus, les images intermédiaires du traitement sont écrites sur le disque pour limiter la consommation de mémoire	Temps de calcul longs. Beaucoup de paramètres (seuls les principaux sont présentés) à tester pour trouver la bonne configuration
Multi-résolution - eCognition	Choix manuel d'un ensemble de cartes de segmentation hiérarchique	Logiciel payant
PYRAM	Cette méthode est simple, n'ayant pas de paramètres à définir. Elle est invariante aux changements linéaires de la dynamique de l'image. La hiérarchie reste la même, avec un axe des échelles modifié en même proportion. Cet algorithme est intéressant pour les images aériennes, car elle peut s'appliquer à des objets de grande taille, tels un village	Sur des images aériennes de grande taille, cette méthode se révèle relativement longue à calculer et coûteuse en mémoire
RSGISlib	Cet algorithme présente de bonnes performances (temps de calcul et gestion de la mémoire). Il peut être appliqué sur de grandes images sans nécessiter de créer une mosaïque de l'image et ne comprend que deux paramètres ce qui permet de l'adapter facilement aux données de précision et d'emprise spatiales diverses	

Conclusion

Le choix d'un algorithme de segmentation pour monter une chaîne de production de l'OCSGE doit répondre aux exigences suivantes :

- Un paramétrage adaptatif
- Une stabilité des résultats

- Une accessibilité suffisante (via SIG, ligne de commande ou code avec une installation facile ou documentation détaillée)
- Pouvoir traiter de grandes images

Il apparaît qu'une segmentation hiérarchique soit plus appropriée pour que l'annotateur puisse jouer avec les différents niveaux de détails afin d'annoter rapidement et avec précision. La segmentation pyram répond aux besoins de l'IGN. Il reste possible de comparer avec d'autres segmentations hiérarchiques pour gagner en temps de traitement mais l'outil implémenté fonctionne bien. Il est paramétrable, fournit des résultats stables, un paramètre de taille de segment permet d'effectuer des opérations de fusion en fin de traitement.

La première opération de segmentation étant satisfaisante en l'état, il convient surtout d'avoir un outil suffisamment performant pour réaliser la deuxième phase, afin que l'exploration des différents niveaux de segmentation soit le plus fluide possible. Il y a un gros intérêt de visualisation dynamique, et de zoom et dé-zoom dans l'image à travers les niveaux de la hiérarchie.

Annexe - A : Plateformes et outils de labellisation

Plateformes propriétaires

Amazon sagemaker ground truth

[lien](#) Outil d'étiquetage de données proposé dans les différents services cloud d'Amazon. Conçu pour être utilisé/intégré avec les autres outils de machine learning et stockage de données proposés par Amazon (AWS).



Exemple de workflow amazon utilisant l'outil Sagemaker ground truth (© Amazon)

L'outil est assez générique et gère la plupart des types de données (images, vidéos, nuages de points 3D, texte). Les interfaces de saisie sont relativement simples. Il y a cependant une option d'automatisation de la saisie avec apprentissage actif d'un modèle deep learning sur un jeu de données. Cette option est disponible pour les tâches de type objet detection (boite englobante + classe) et segmentation d'objet. Les corrections par rapport à une prédiction automatique sont faites avec des outils de saisie simples (brush, polygone, déplacement de sommets). Il n'y a pas d'exemples montrant l'efficacité de la méthode sur des données de types aériennes/spatiale et une segmentation de type OCS (ie pas une segmentation d'objet avant/arrière-plan).

Cloud Annotation (IBM)

[lien](#). Outils de saisie pour labellisation image et détection objet (bounding box) d'IBM. Open-source pour la partie client mais fonctionne seulement sur des objets/images stockés sur le cloud d'IBM. Propose également d'entraîner des modèles, toujours sur cloud IBM, pour automatiser la labellisation et l'amélioration du modèle de détection.

Hasty.ai

[\(lien\)](#). Plateforme de labellisation pour les données images et plutôt dédiée à la détection et segmentation d'objet. Offre des fonctionnalités de saisie avec plusieurs niveaux d'automatisation. Soit de la saisie assistée avec des outils de type contour/grab cut ou detxr ou bien de la saisie plus automatique avec des pré-détection et pré-segmentation à partir de modèle deep learning entraîné de façon interactive. La partie de gestion de données et projet (rôles) est aussi assez complète.

Labelbox

[Labelbox](#) Plateforme de labellisation générique avec une part importante d'outils pour la labellisation image. Propose une offre complète allant des outils de labellisation à la sous-traitance de labellisation (mise en relation avec des annotateurs) et comprend une part d'entraînement de modèle possible. Il s'agit

d'une des plateformes les plus présentes sur le marché. Possède des fonctionnalités compatibles avec l'annotation de données géographiques (flux de type wms) et affichage d'images multi-spectrales. Possède des outils pour la labellisation de segmentation sémantique basé sur les superpixels.

Playment.io

<https://playment.io/gt-studio>.

Plateforme de labellisation de données orienté images/vidéo et multi-sensor. Plutôt orientée sur le marché de l'annotation de données pour les véhicules autonomes. Possède des outils d'annotations images de types éditions raster ainsi que des outils d'automatisation pour la labellisation de type instance ou segmentation. Par contre ne propose pas de saisie sur des segmentations de types superpixels et n'a pas de fonctionnalités dédiées aux données géomatiques.

Pointly

<https://pointly.ai/>

Plateforme de labellisation orientée sur la labellisation de données Lidar et nuages de points.

Rectalabel :

<https://rectlabel.com/> Application de bureau pour MAC permettant la labellisation image pour la détection d'objet, la segmentation sémantique et l'éditions de keypoints/marker (exemple sur les personnes). Comprend juste des fonctionnalités de saisie et lecture de fichiers images dans un dossier pas de fonctionnalités avancées de collaboration, contrôles qualité et gestion de projet/tâches.

Superannotate

Plateforme de labellisation de données plutôt spécialisée dans l'annotation d'image et de vidéo. Son principal intérêt est le fait de proposer un outil de saisie de données de segmentation sémantique qui semble performant/efficient et relativement compatible avec les données géomatiques. Ce dernier est basé sur une segmentation hiérarchique et propose une interface graphique relativement bien pensée pour cette tâche. Elle propose aussi une version desktop de son outil d'annotation mais sans la partie segmentation qui dépend de calcul sur un serveur, la version plateforme peut être déployée sur une infra propre.

Supervise.ly

([lien](#)) Plateforme de labellisation complète pour la saisie de données sur images, nuages de points 3D et vidéos. Offre des fonctionnalités d'automatisation de la saisie mais principalement par la détection et segmentation d'objet et non pour la segmentation sémantique/panoptique. Offre un SDK et un système de plugins assez complet. L'automatisation comprend la possibilité de configurer et entraîner des modèles d'apprentissage profond. Les fonctionnalités de gestion de projet et des droits sont assez poussées.

Toutefois cet outil n'offre pas d'outils de saisie de type superpixel/segmentation ni de fonctionnalités particulières pour les données images aériennes ou spatiales.

Outils / plateforme open-source

CVAT

Un des premiers outils de labellisation open-source et offrant des fonctionnalités de saisie assez complètes. Il est toutefois plutôt orienté pour la saisie d'objet/instance et la thématique des véhicules autonomes. Fonctionné en mode client/serveur avec au niveau technique python et Django en technologie côté serveur et du javascript côté client.

Label Studio

Plateforme de labellisation fonctionnant sur un double modèle. Une partie open-source et une offre commerciale propriétaire. La plateforme open-source est plus limitée que la partie commerciale, notamment au niveau de la gestion des droits utilisateurs et des outils de contrôles qualité. Par contre la communauté open-source semble assez active pour le moment. Toutefois, il n'y a pas de fonctionnalité de labellisation de segmentation sémantique basée sur une segmentation image pour l'instant (fonctionnalité [à l'étude](#)). L'intérêt de la plateforme est donc à ce jour d'offrir une base de développement assez robuste et complète qui peut être adaptée pour un besoin propre. La plateforme comprend plusieurs projets relativement indépendants :

- label-studio : partie serveur de la plateforme de labellisation, gestion de données.
- [label-studio-frontend](#) : client web de labellisation basé sur React et javascript.
- [Data-manager](#) : client web de gestion de données labellisées utilisant le serveur label-studio et des composants du frontend.
- [label-studio-converter](#) : utilitaire/librairie de conversion de format entre données labellisées et outil/framework IA.

MakeSense.ai

[lien](#) Outil de saisie, orienté web pour la détection et segmentation d'objet. Il s'agit juste d'une interface de saisie web fonctionnant sur des images en local et il n'y a donc pas de fonctionnalités de gestion de projets et éditions collaborative. Bien que fonctionnant côté client l'outil propose des fonctionnalités de segmentation automatique en faisant tourner des modèles de segmentation pré-entraînés sur le navigateur en utilisant le framework tensorflow.js.

Dafne : Deep Anatomical Federated Network

([lien](#)) Plateforme de labellisation et partage de modèles deep learning pour la segmentation d'image médicale MRI (IRM en français : image à résonance magnétique). Développée en python et open-source. Compris à la fois une partie serveur et un client pour la labellisation de donnée.

<https://dafne.network/tutorials>

Liste autres outils open-sources :

- [Lost](https://github.com/l3p-cv) (https://github.com/l3p-cv)
- [Pixie](#)
- [VoTT](#) (Visual Object Tagging Tool) de Microsoft
- [Django-labeller](https://github.com/Britefury/django-labeller) <https://github.com/Britefury/django-labeller>
- <https://github.com/scalabel>
- <https://github.com/medtagger/MedTagger>
- <https://github.com/randxie/react-pixel-annotation-tool>
- Sentinel-hub classification app_
(<https://github.com/sentinel-hub/classification-app-frontend>)

Annexe - B : Fiches fonctionnalités de saisie image

Table matière à insérer

- Annotation / labellisation de classe image
- Saisie d'objet / boites englobantes image
- Saisie vecteur (polygone, lignes) sur image
- Saisie avec outils d'éditeurs raster
- Saisie par classification pixel d'image
- Saisie avec segmentation d'image.

Annotation / labellisation de classe image

Description

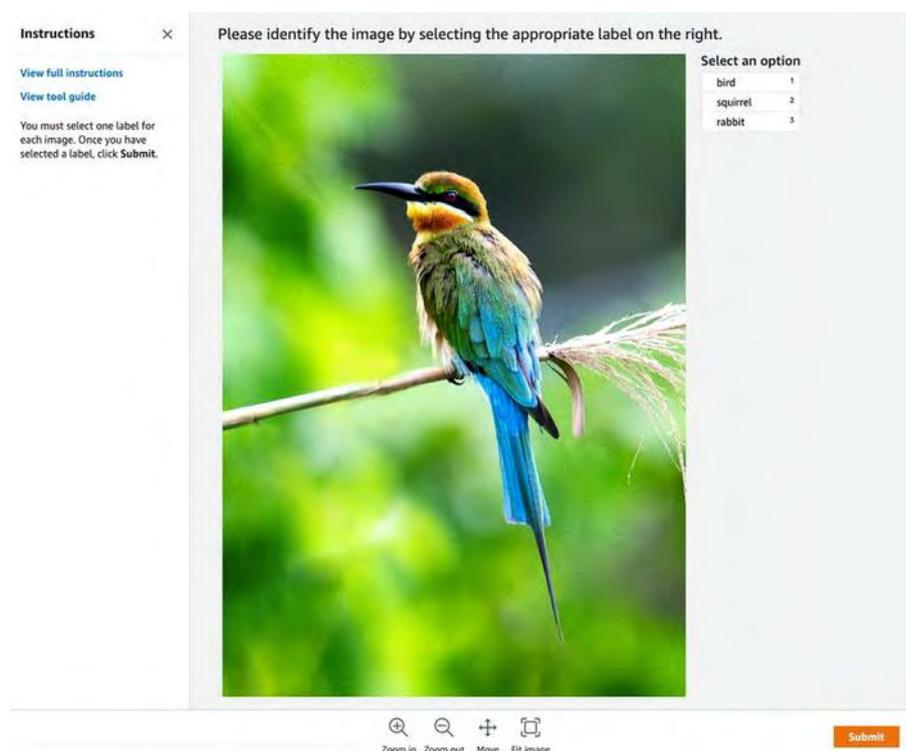
Choix d'une/plusieurs classes par images.

De plus les classes peuvent également être ordonnées de façon **hiérarchique** (arbre de classes plutôt qu'une simple liste) et la labellisation peut être **simple ou multi-label**. Enfin il peut s'agir d'une interface indépendante pour une tâche de classification ou bien une **interface complémentaire à d'autres tâches** comme la segmentation. En effet cela peut être ajouté afin d'informer sur la qualité des images à labelliser et affecter des label/tag qualité de type : "Floue / Pluie / neige / nuages / sombre"

UX

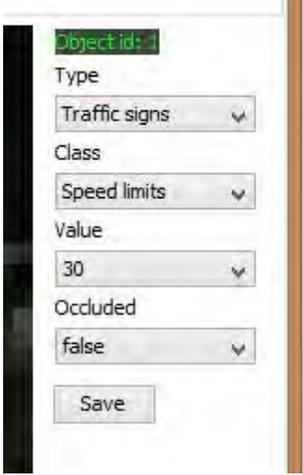
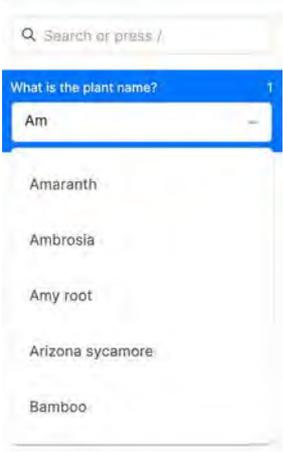
Les interfaces associées à cette tâche sont souvent simples et constituées d'éléments de type choix multiples (radio button, checklist).

Pour les listes de classes courtes et pour rendre la saisie plus rapide il peut y avoir un bouton par classe, associé généralement à un raccourci clavier. Ce dernier peut être indiqué au niveau du label du bouton (chiffre, touche associée après le nom de la classe ou bien lettre soulignée pour indiquer la touche associée à la classe).

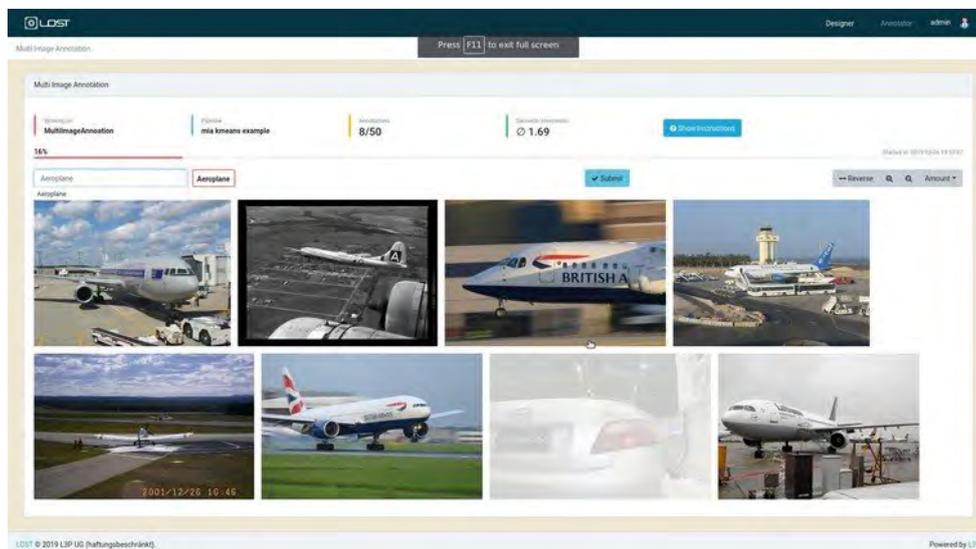


Exemple interface [classification image amazon sagemaker](#) (© Amazon)

Pour les classes nombreuses ou hiérarchiques, on peut voir soit des boutons/listes par niveaux ou bien un choix via listes déroulantes/texte éditable proposant aussi une complétion automatique.

	
<p>Classe hiérarchique (© pixie)</p>	<p>Complétion automatique (© labelbox)</p>

Options : Choix possible de découper un job multi-label en n job de labellisation binaire. Ou bien d'afficher/labelliser plusieurs images en mêmes temps. Dans le cas de classification binaire cela peut être plus efficace et rapide.



Exemple de labellisation multi-image (© [lost](#))

Automatisation : Cette tâche de labellisation d'image peut être convertie en tâche semi-automatique de contrôle lorsque l'on dispose de données déjà labellisées et d'un premier modèle de prédiction. L'interface propose alors des labels pré-calculés à modifier/valider, soit en pré-affichant les labels dans l'interface de saisie soit en demandant à l'opérateur de valider ou non la classification et en ouvrant l'interface de labellisation en cas d'erreur du modèle.

Liens

- <https://labelbox.com/blog/improved-search-in-the-new-editor/>
- Microsoft "label image and document"
- Kili-technology [image interface classification](#)

Saisie objet / boîte englobante sur image

Description

Boîte englobante sur les objets d'intérêt de l'image. Saisie de la classe des objets

Il s'agit souvent des saisir les objets en premier plan dans l'images et qui constituent les sujets principaux de celles-ci. Le nombre d'objet à saisir peut varier de 2 ou 3 à presque une centaine dans le cas de scènes complexes (identification de personnes dans une foules, de voiture dans un trafic congestionné etc..)

UX

Les interfaces de saisie sont assez variées et proposent souvent des améliorations par rapport à l'interface basique consistant à saisir une boîte englobante en saisissant une diagonal (coin haut/gauche <-> coin bas/droit).



Interface de saisie boîte englobante (© kili-technology) affichage du réticule.

Une boîte englobante saisie est la plupart du temps modifiable par déplacements des bords/Sommet.

Les coins de la boîte englobante n'étant souvent pas sur l'objet il est assez difficile d'estimer de façon précise leurs positions sans aide supplémentaire. Pour cela certaines interfaces affichent un réticule lors de la saisie.

En termes de saisie d'objet il a été noté que la saisie de boîte englobante pouvait être moins efficaces que la saisie de 4 points extrêmes sur les objets.



illustration "[Extreme clicking for efficient object annotation](#)"

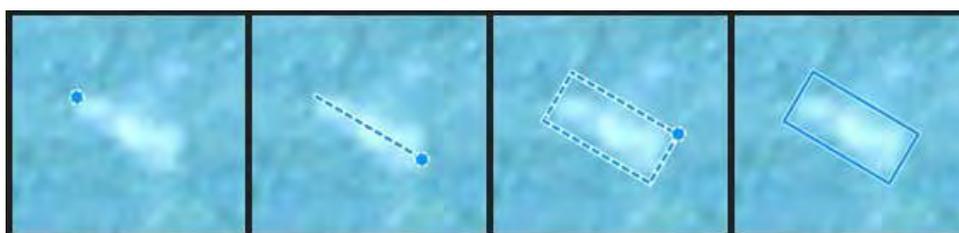
Extreme clicking :

*"It is much easier to label the extreme points of an object than the bounding box. To label a bounding box, you must eye-ball where the corner is that aligns with the two most extreme points. Usually the annotator must adjust the bounding box afterwards to line up with the extreme points anyways. The ExtremeNet paper estimates that **labeling extreme points can take almost a fifth the amount of time it takes to label bounding boxes**"*

Options/ Variantes

En plus de la modification de type resize via les bords ou sommet certaines interfaces proposent aussi des fonctionnalités de translation (rotation possible mais pas vu en labellisation non sig).

Ce mode de saisie peut aussi être adapté à d'autres formes géométriques simples comme les cercles ou ellipses. Des cas d'utilisation peuvent être la saisie de fruit ou arbres/rond points dans des cas géomatiques. Un exemple dans le cas de la saisie pour la reconnaissance en milieu aquatique avec "biigle.de". Plus d'exemples sont décrits par HumanInTheLoop (<https://humansintheloop.org/types-of-image-annotation/>)



[Saisie boîte orientée](#) (© [biigle.de](#))

Automatisation : Possible en interfaçant un modèle de détection d'objet de type detectron 2. Par exemple voir l'outil [anno-mage](#).

Liens :

- [Kili technology bounding box](#)
- Microsoft [VoTT](#)
- Make-sense.ai (<https://github.com/SkalskiP/make-sense>)
- Outil d'IBM <https://cloud.annotations.ai/>
- LabelMe <https://github.com/wkentaro/labelme>

Saisie vecteur (polygones, lignes) sur image

Description

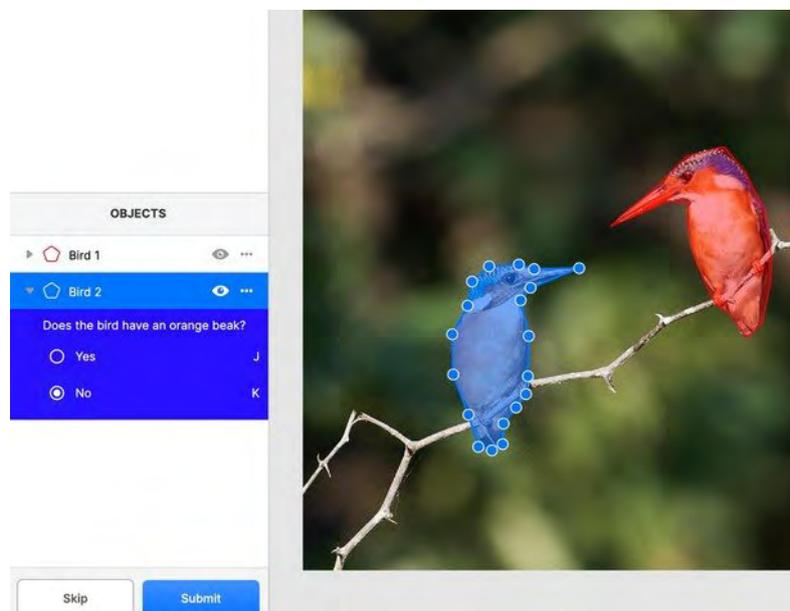
Saisie de polygones/vecteurs suivant les contours d'un objet.

La plupart des outils proposent l'édition de polygones mais certains proposent également la saisie d'autre type de géométrie comme les points ou bien les polygones. Dans le domaine géomatique, en image terrestre ou aérienne les polygones peuvent servir pour la saisie des bords de routes ou bien de marquage routier.

UX

Les interfaces proposées sont assez simples et très similaires à celles qui existent en géomatique dans les SIG ou les bibliothèques web de type OpenLayer.

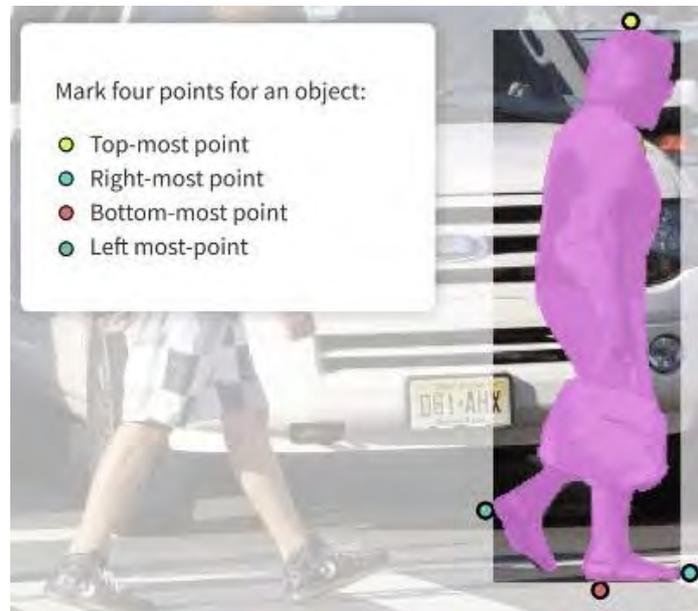
- Autocomplétion / arrêt saisie
- Edition (ajout /déplace/supprime sommets)
- Raccourcis (suppr dernier segment / zoom/ quitter interface saisie)



Exemple de saisie de polygone (© labelbox)

Automatisation : Une première manière de rendre l'outil plus rapide est via l'utilisation de mode de saisie **"free-hand"** et/ou **l'utilisation de lasso-magnétique**. L'utilisateur suit alors le contour de façon grossière à main levée et l'outil s'accroche à la limite objet (gradient/détection de contour) le plus proche. Le nombre de sommets peut être un paramètre de saisie si on ne souhaite pas un sommet par pixel (Pour plus de détails voir par exemple une explication de ce mode de saisie dans l'outil de dessin krita [magnetic_select](#))

La deuxième méthode est de demander à l'utilisateur de saisir une boîte englobante ou **les points extrêmes** puis de lancer un algorithme de segmentation (deep ou non) séparant objet et arrière plan (background/foreground)



Saisie “intelligente” de polygone via la méthodologie Dextr.

Liens :

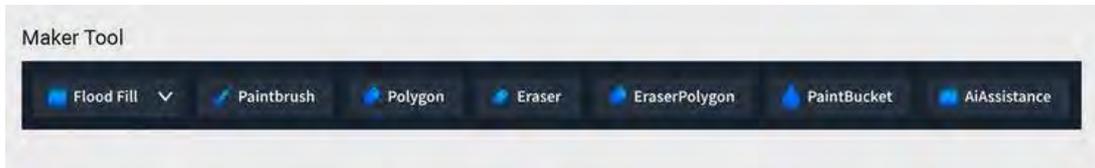
- Labelbox pen tool (<https://labelbox.com/blog/introducing-image-segmentation/>)
- Example saisie freehand + accroche magnetique : <https://github.com/PyIFT/livewire-gui>
- [Smart polygon GrabCut harty.ai](https://harty.ai)

Saisie avec outils d'édition raster/image

Description:

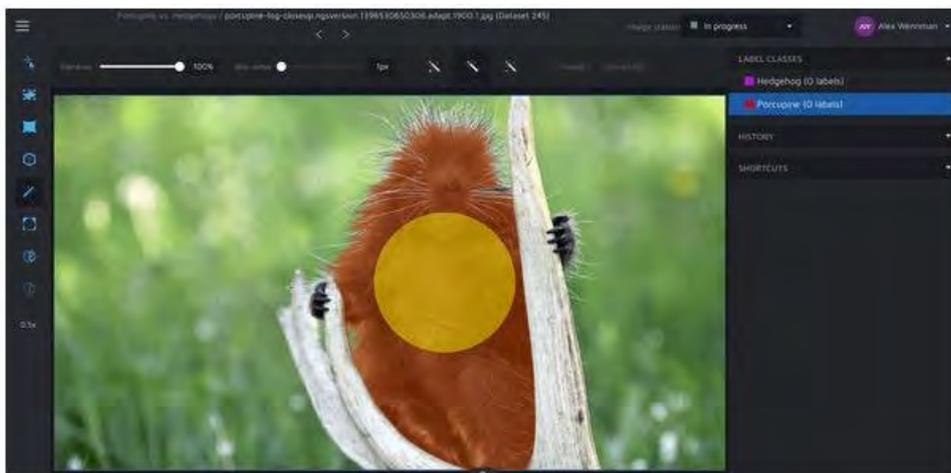
Saisie des labels pour segmentation sémantiques à partir d'outils d'édition raster/images standard. Similaire aux outils présents dans Paint/photoshop/gimp/Krita etc..

UX



[Barre d'outils d'édition images playment \(©\)](#)

- Floodfill/ magicwand/ pipette magique : sélectionne région de "couleur" homogène à partir d'un échantillon (point ou cercle de taille variable)



Saisie avec [pipette magique dans hardy.ai \(©\)](#)

- Brush / pinceau : colorie pixel avec largeur définie (à la paint/gimp)



[Utilisation de l'outil brush \(mode erase\) pour editer une segmentation © django labeller](#)

- Eraser / gomme : similaire à brush mais efface les annotations.
- Pot de peinture (bucker) : remplis une zone non labellisée avec la classe sélectionnée.
- Gestion de couche/calques (comme photoshop/gimp)

Automatisation : Peu d'automatisation sur ces outils. Les fonctionnalités sont globalement copiées depuis les logiciels d'éditions photo/images existants. Ces outils peuvent être utilisés pour retoucher le résultat d'une segmentation calculée de façon automatique.

Liens :

- Django labeller <https://github.com/Britefury/django-labeller>
- Superannotate outils pixel : <https://doc.superannotate.com/docs/pixel>
- Matlab/mathworks [label-pixels-for-semantic-segmentation](#)

Saisie avec classification d'image au pixel

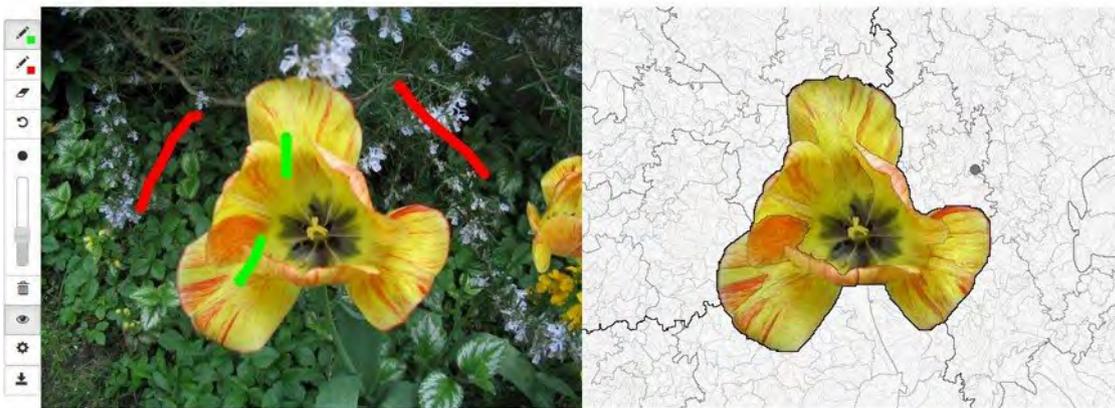
Description:

Classification au pixel d'une image à partir d'interaction/choix utilisateur.

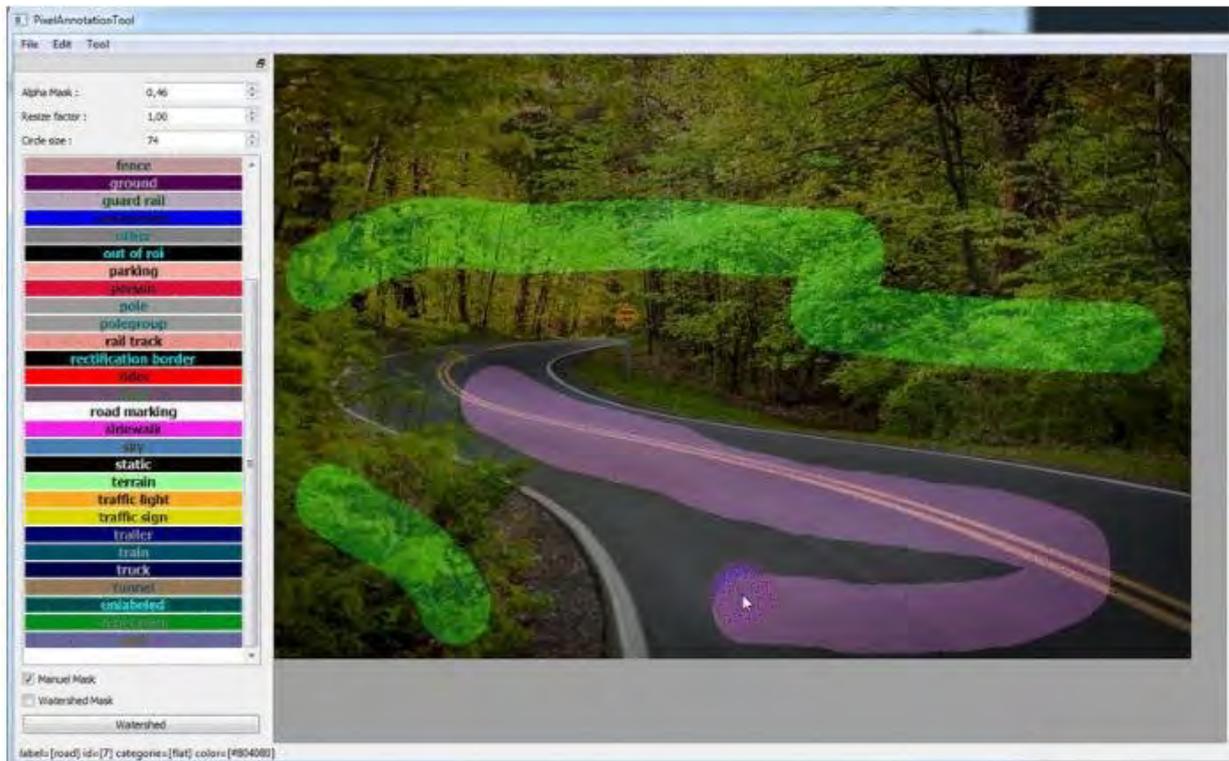
Utilise généralement des méthodologies machine learning mais peut aussi utiliser des algorithmes deep learning. Un exemple d'utilisation en données de télédétection est [doodle_labeller](#)

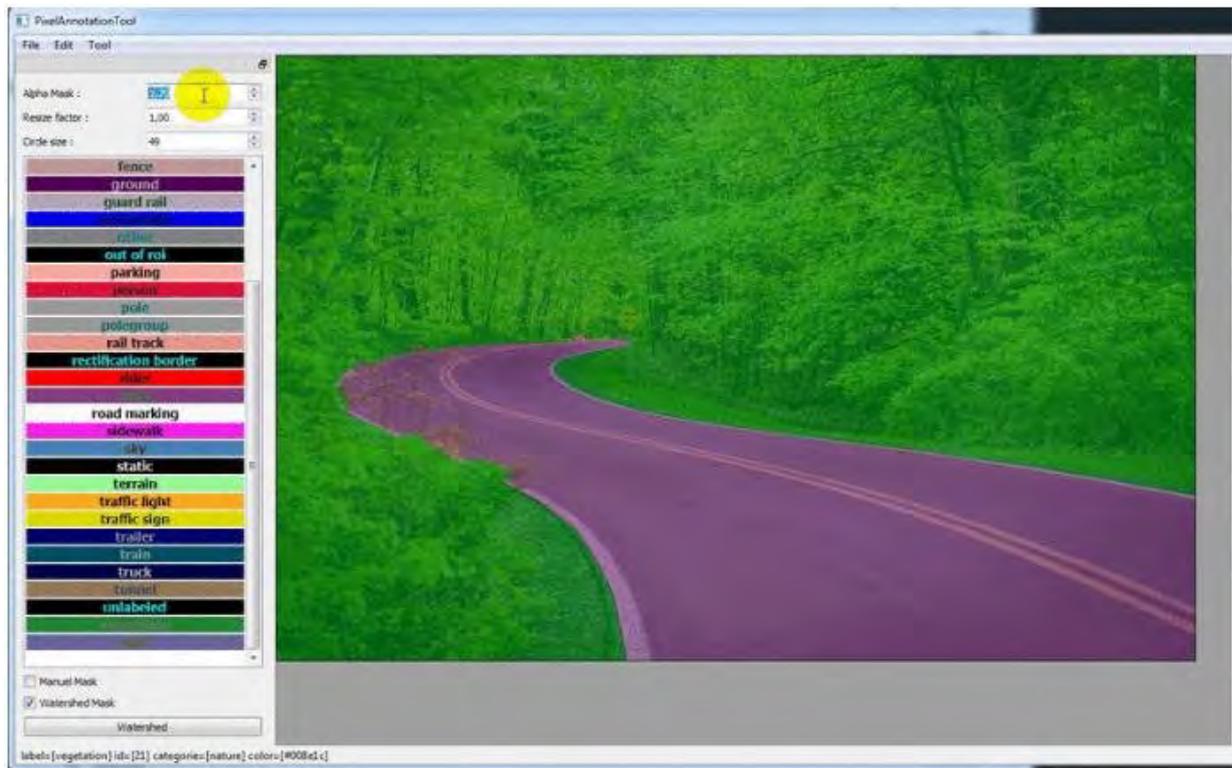
UX

L'interface dépend de la méthodologie de classification utilisée. Cela consiste essentiellement à tracer des traits (strokes/doodle) sur les différentes classes qui servent comme exemples pour l'apprentissage machine. Afin d'avoir une certaine continuité de la classification la méthode peut utiliser une pré segmentation de l'image.



Segmentation par marked watershed (© <https://perso.esiee.fr/~perretb/ISeg/>)





Exemple de segmentation image avec [PixelAnnotationTool](https://github.com/abreheret/PixelAnnotationTool). ©

Option/variantes: Les variantes consistent essentiellement aux choix de la méthodologie de classification à utiliser dans la méthode. La plupart des outils utilise une méthodologie de type marked watershed mais d'autres solutions, dont deep learning sont possibles. L'interface s'adapte alors aux contraintes de l'interfaces (saisie de lignes/points etc..)

Automatisation : La méthode étant déjà en partie automatisé il y a peu d'automatisation supplémentaire possible.

Liens :

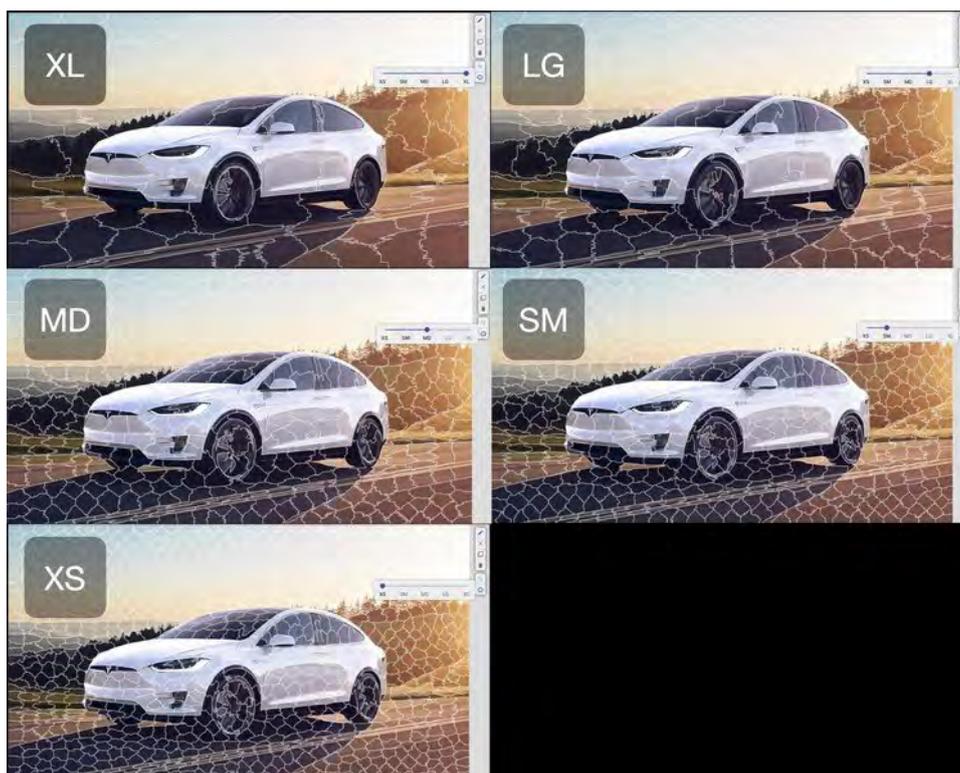
- PixelAnnotationTool <https://github.com/abreheret/PixelAnnotationTool>
- <https://reactjsexample.com/pixel-annotation-tool-written-in-react-and-opencv-js/>
- Doodle labeller : https://github.com/dbuscombe-usgs/doodle_labeller

Saisie raster à partir de segmentation image

Description

Sélectionne les régions de l'image appartenant à une classe à partir d'une segmentation image déjà faite.

Options/ Variantes : La plus grande différence entre les solutions existantes vient de la configuration de la segmentation à utiliser. Est-elle préfixée ou bien l'utilisateur peut-il choisir plusieurs tailles possibles :



différents niveaux de segmentation pour la labellisation (© kili-technology)

Liens

<https://kili-technology.com/blog/what-is-the-best-segmentation-tool/>

<https://github.com/kyamagu/js-segment-annotator>

Annexe - C : Détails outils et algorithmes de segmentation

- Approfondissements (superpixels, segmentation hiérarchique)
- Segmentation Felzenszwalb
- Segmentation Quickshift
- SLIC : Simple Linear Iterative Clustering
- Watershed & Compact Watershed
- Mean shift et variante Large Scale Mean-Shift
- Segmentation Multi-resolution (Batz) logiciel ECognition
- Segmentation avec un processus d'élimination itérative de Shepherd et al. (RSGISlib)
- Segmentation Multi-échelle Guigues (PYRAM - IGN)

Algorithmes superpixels

Formalisés par [Ren et Malik, 2003](#), leur philosophie dérive de la [théorie de la Gestalt](#). C'est une approche qui s'intéresse à la façon dont l'être humain identifie des patterns, des relations entre les objets et prend en compte le contexte. Les superpixels s'attachent donc à prendre en compte des critères de proximité, similarité, continuité et l'approche top-down afin de segmenter l'image d'une manière plus proche de la vision humaine.

Principe : l'algorithme de Ren et Malik découle du Normalized Cuts formulé par [J. Shi et J. Malik, 2000](#) auquel ils ont ajouté une contrainte de proximité. Il s'agit d'une segmentation par région qui partitionne récursivement un graphe donné en utilisant les indices suivants : similarité de luminosité et de texture intra/inter-région, prise en compte de l'énergie de contour intra/inter-région et de la continuité curviligne pour avoir des frontières lisses. Elle minimise une fonction de coût définie sur les bords de la segmentation. Il en résulte une bonne approximation d'une segmentation qui serait faite par un humain.

Propriétés des superpixels :

- **Efficacité de calcul** : en passant des pixels aux superpixels, réduit la complexité des images et le nombre de primitives. Temps de calcul presque linéaire par rapport au nombre de pixels de l'image [[Felzenszwalb et Huttenlocher, 2004](#)]
- **Représentation plus naturelle** : plus proche de la vision humaine que les pixels qui sont une discrétisation de l'image
- **Représentation significative** : les pixels d'un superpixel forment une unité cohérente, ils sont homogènes en termes de couleur et de texture
- **Segmentation presque complète** : très peu de perte de structures de l'image lors du passage des pixels aux superpixels

Tout algorithme de superpixels est donc une sur-segmentation avec des segments de taille régulière et similaire, qui sont distribués uniformément dans l'image.

Segmentation hiérarchique

Il s'agit d'un ensemble de régions structurées hiérarchiquement. Il peut être désigné par la notion d'arbre également. Ci-dessous, une définition plus formelle d'un ensemble multi-échelle :

« En termes de partitions, une séquence $(P_\lambda)_{\lambda \in \mathbb{R}^+}$ de partitions peut donc être considérée comme une séquence multi-échelles de paramètre d'échelle λ , si pour tout $\lambda_1 < \lambda_2$, P_{λ_1} est systématiquement une sur-partition de P_{λ_2} , c'est-à-dire s'obtient en redécoupant certains des ensembles de P_{λ_2} . $(P_\lambda)_{\lambda \in \mathbb{R}^+}$ constitue alors une séquence de partitions emboîtées, de finesse décroissante avec λ . » [Guigues et al., 2003]

Et de hiérarchie indicée :

« Un couple (H, λ^+) où H est une hiérarchie de régions (un système d'ensembles deux à deux disjoints ou emboîtés) et λ^+ est une fonction de H dans \mathbb{R}^+ qui indique pour chaque région r la plus petite échelle à partir de laquelle r appartient à une des partitions de P . Nous avons appelé cette échelle caractéristique l'échelle d'apparition de r . » [Guigues et al., 2003]

Algorithmes existants :

Le quad-tree qui définit une hiérarchie avec pour racine l'image entière. Celle-ci est découpée entre quatre parties, qui le sont à leur tour récursivement jusque les parties soient toutes décrétées homogènes.

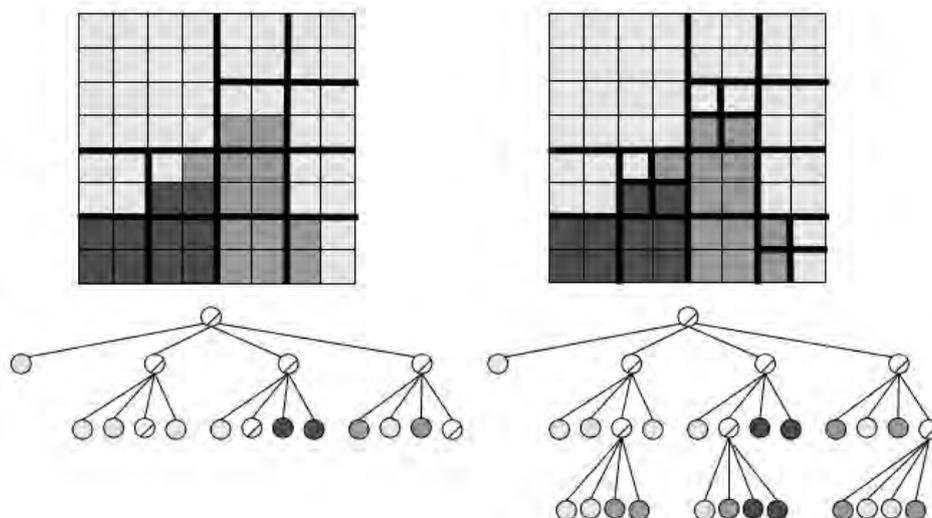


Image tirée d'un cours de segmentation de l'école Télécom Physique Strasbourg

Apport annotation : il en ressort une segmentation de régions carrées, cette structure rigide n'est pas très intéressante pour notre étude.

Felzenszwalb

#region-growing #graph-based #superpixels

Référence : Felzenszwalb, Pedro F., et Daniel P. Huttenlocher. « Efficient Graph-Based Image Segmentation ». *International Journal of Computer Vision* 59, n° 2 (septembre 2004): 167-81. <https://doi.org/10.1023/B:VISI.0000022288.19776.77>.

Principe : Cet algorithme de fusion de régions se fonde sur les méthodes des graphes pour regrouper les points similaires en utilisant un arbre à portée minimale. Chaque segment, ou superpixel, est l'arbre le plus court des pixels qui le composent. Son apport est de prendre en compte des caractéristiques de l'image non locales qui sont importantes d'un point de vue perceptif pour déterminer les différentes régions. Les pixels sont regroupés selon une distance euclidienne utilisée dans l'espace colorimétrique.

Fonctionnement : L'image est traitée comme un graphe non orienté dans lequel chaque pixel correspond à un nœud. Certains pixels voisins sont reliés par des bords dont le poids indique la dissimilarité entre ces pixels. Il s'agit d'une fusion ascendante de pixels en région. Chaque région est comparée à la région voisine dont le poids du bord est minimal. Si la différence d'intensité à travers la frontière entre les deux régions est importante par rapport aux différences d'intensité à l'intérieur d'au moins une des régions, alors elles sont fusionnées.

L'algorithme Felzenszwalb est utilisé avec deux types de graphes différents :

- la grille de l'image permet de définir un voisinage local entre les pixels de l'image, et mesure la différence d'intensité (ou de couleur) entre chaque paire de voisins.
- Chaque pixel est associé à un point dans un espace caractéristique à 5 dimensions (x,y,r,v,b) combinant l'emplacement et la couleur du pixel. Les arêtes du graphes relient ainsi les points proches dans l'espace des caractéristiques. Ce deuxième type de graphe met en évidence des caractéristiques plus globales de l'image d'un point de vue perception.

Paramètres implémentation Scikit-image :

- **scale** : niveau d'observation. Une échelle de plus en plus petite signifie des segments de plus en plus petits. Le nombre de segments produits et leur taille sont donc contrôlés indirectement à travers le paramètre scale sachant que le contraste local fait également varier fortement leur taille.
- **sigma** : diamètre du noyau gaussien utilisé dans le pré-traitement pour lisser l'image avant la segmentation
- **minsize** : taille minimal des segments appliquée lors du post-traitement

Complexité algorithmique : $O(N \log N)$ pour N pixels. Presque linéaire par rapport au nombre d'arêtes du graphe. Il peut fonctionner en pratique à des vitesses vidéo.

Avantages : préserve les détails dans les régions de l'image à faible variabilité et l'ignore dans les régions à forte variabilité. Suffisamment rapide pour pouvoir être utilisé sur des images à grande échelle (Ratan et al. 1999).

Inconvénients : pas moyen de contrôler le nombre ou la compacité des superpixels directement ([R. Achanta et al., 2010](#)).

Quickshift

#clustering #mode-seeking

Référence : AVedaldi, Andrea, et Stefano Soatto. « Quick Shift and Kernel Methods for Mode Seeking ». In *Computer Vision – ECCV 2008*, édité par David Forsyth, Philip Torr, et Andrew Zisserman, 705-18. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008. [A. Vedaldi et S. Soatto, 2008. https://doi.org/10.1007/978-3-540-88693-8_52](https://doi.org/10.1007/978-3-540-88693-8_52).

Contexte : Dans la littérature des algorithmes de clustering, on a montré que l'algorithme medoid shift est plus rapide que [mean shift](#) dans le regroupement de pixels si la distance utilisée est euclidienne. Or, l'un des inconvénients du medoid shift est qu'il ne parvient pas à regrouper les pixels de même mode, ce qui entraîne une sur-segmentation. C'est dans ce contexte que ces auteurs ont développé l'algorithme quickshift qui donne de meilleures segmentations que mean shift et medoid shift, avec un temps de calcul plus court.

Principe : Quickshift est un algorithme de clustering. Il cherche les modes d'énergie et regroupe un pixel avec son voisin le plus proche qui a le niveau d'énergie le plus élevé. Il équilibre entre sur-segmentation et sous-segmentation.

Fonctionnement : Pour cela, l'algorithme commence par calculer une estimation de la densité de Parzen $P(x)$. Puis il déplace chaque point dans l'espace des caractéristiques des pixels vers le point le plus proche dans un voisinage τ qui augmente $P(x)$. Cet algorithme de clustering aboutit à un arbre qui relie tous les points. Le paramètre τ est le seuil qui permet de casser les branches de l'arbre plus longues que τ afin de récupérer les modes [[A. Vedaldi et S. Soatto, 2008](#)]. C'est τ qui permet l'équilibre de la segmentation.

Paramètres implémentation Scikit-image :

- **kernel_size** : largeur du noyau gaussien utilisée pour lisser la densité de l'échantillon. Une grande valeur implique moins de clusters.
- **max_dist** : point de coupure pour les distances de données. Une grande valeur implique moins de clusters.
- **ratio** : équilibre la proximité de l'espace colorimétrique et la proximité de l'espace image. Des valeurs plus élevées donnent plus de poids à l'espace colorimétrique.

Complexité algorithme : $O(dN^2)$ pour N points et d la dimension de N .

Avantages : L'algorithme est simple, rapide et permet d'équilibrer entre sous et sur-segmentation des modes. Plus rapide que le Mean-shift.

Inconvénients : pas moyen de contrôler le nombre ou la taille des superpixels directement ([R. Achanta et al., 2010](#)).

SLIC - Simple Linear Iterative Clustering

#superpixel #clustering #gradient-ascent-based

Référence : Achanta, Radhakrishna, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, et Sabine Süsstrunk. « SLIC superpixels ». *Technical report, EPFL*, 19 juin 2010.



Fig. 1. Image segmented using our algorithm into superpixels of (approximate) size 64, 256, and 1024 pixels. The superpixels are compact, uniform in size, and adhere well to region boundaries.

Achanta et al. 2010

Principe : Cet algorithme repose sur le principe des k-means. Des centres de clusters sont initialisés à partir d'une grille uniforme de l'image. Chaque cluster va incorporer les pixels de son voisinage en fonction d'une distance spatiale et d'une distance spectrale. Un paramètre de compacité permet de favoriser l'une ou l'autre de ces distances.

Fonctionnement : Une grille régulière d'intervalle S (racine du nombre de pixels divisé par le nombre de superpixels) est définie. Les centres des clusters sont initialisés au gradient le plus faible dans un voisinage $n \times n$ des centres de chaque carreau de cette grille. Ensuite, à partir d'un voisinage $2S \times 2S$ de chaque centre, on associe de manière itérative les pixels au centre selon une mesure de distance. Celle-ci prend en compte la somme de la distance dans l'espace couleur CIELAB et de la distance dans le plan xy normalisée par l'intervalle de la grille. Cette mesure permet un bon équilibre entre la similarité des couleurs et la proximité spatiale.

Puis on recalcule le centre de cluster jusqu'à convergence ainsi que l'erreur (distance entre les centres précédents et les centres recalculés). Dans [R. Achanta et al., 2010](#), 4 à 10 itérations suffisent. Enfin, on applique la connectivité en labellisant de nouveau les segments disjoints selon l'étiquette du plus grand cluster voisin. Cette étape finale prend moins de 10% du temps total de l'algorithme, elle est de complexité $O(N)$.

Paramètres implémentation Scikit image :

- ***n_segments*** : nombre (approximatif) de centres pour les kmeans / de segments générés en sortie.

- **compactness** : faire un compromis entre la similarité des couleurs et la proximité spatiale (comme le paramètre ratio de l'algorithme Quickshift). Des valeurs élevées privilégient l'espace, ce qui donne des formes de superpixels plus carrées. Cela dépend néanmoins du contraste de l'image et des objets qu'elle contient. Il est conseillé de tester les valeurs 0,01, 0,1, 1, 10, 100, avant d'affiner autour d'une valeur choisie.
- **sigma** : largeur du noyau gaussien utilisé pour le lissage de chaque bande de l'image en pré-traitement. Sigma = 0 signifie qu'il n'y a pas de lissage

Complexité algorithmique : $O(N)$ pour N pixels. On calcule la distance de chaque pixel à un maximum de 8 centres de clusters.

Avantages : cet algorithme produit des segments compacts et presque uniformes à un faible coût. Il peut être codé avec un seul paramètre : le nombre de pixels.

Inconvénients : faible qualité de la segmentation. L'utilisateur définit le paramètre de compacité qui est donc le même pour l'ensemble de l'image. SLIC a donc tendance à produire des superpixels de taille régulière dans les régions lisses et d'autres de taille très irrégulière dans les régions texturées. Les aspects de l'image ne sont pas assez pris en compte.

Watershed

#region growing #gradient-ascent-based

Principe : Introduit par [S. Beucher and C. Lantuéj en 1979](#), l'algorithme watershed considère l'image en entrée comme un relief topographique dont on simule l'inondation. Les valeurs des pixels sont donc des mesures de l'élévation du relief. On obtient en sortie les frontières qui sont les lignes de partage des eaux entre deux bassins versants. Un bassin versant est une région dans laquelle toutes les eaux de surface s'écoulent vers un exutoire. On peut construire une hiérarchie de bassins versants, englobant d'autres bassins-versants et variant ainsi en échelle. Cet algorithme repose généralement sur l'utilisation d'un modèle numérique de terrain (MNT). L'un des algorithmes des plus courants et des plus utilisés est l'algorithme de [S. Beucher and F. Meyer, 1993](#).



*Illustration en une dimension tirée d'un cours de segmentation de l'école
Télécom Physique Strasbourg*

r.watershed de Grass

#region growing #gradient-ascent-based

Références : cf. <https://grass.osgeo.org/grass78/manuals/r.watershed.html>

Principe : Algorithme de recherche au moindre coût qui génère un ensemble de cartes indiquant : l'accumulation des flux, la direction du drainage et l'emplacement des cours d'eau et des bassins versants.

Fonctionnement : L'écoulement de surface peut être calculé de deux façons :

- La direction d'écoulement unique (SFD, D8)
- La direction d'écoulement multiple (MFD) : distribution proportionnelle dans toutes les cellules voisines d'élévation inférieure, avec la pente (chemin le moins coûteux) entre la cellule centrale et sa voisine comme facteur de pondération. Un facteur de convergence entre 1 et 10, permet de faire converger l'accumulation de flux plus ou moins fortement. La valeur recommandée est 5 (Holmgren, 1994).

Paramètres :

- Fournir un MNT en entrée
- Taille minimale d'un bassin versant : ainsi, pour délimiter un grand bassin versant drainant une bonne partie de la région d'étude, il sera conseillé de fixer le seuil à 10000 pixels à une résolution de 30mètres.¹⁴

Avantages : précis dans les zones de faible pentes, capable de reproduire l'emplacement des cours d'eau dans les grandes vallées, même si le couvert forestier est important.

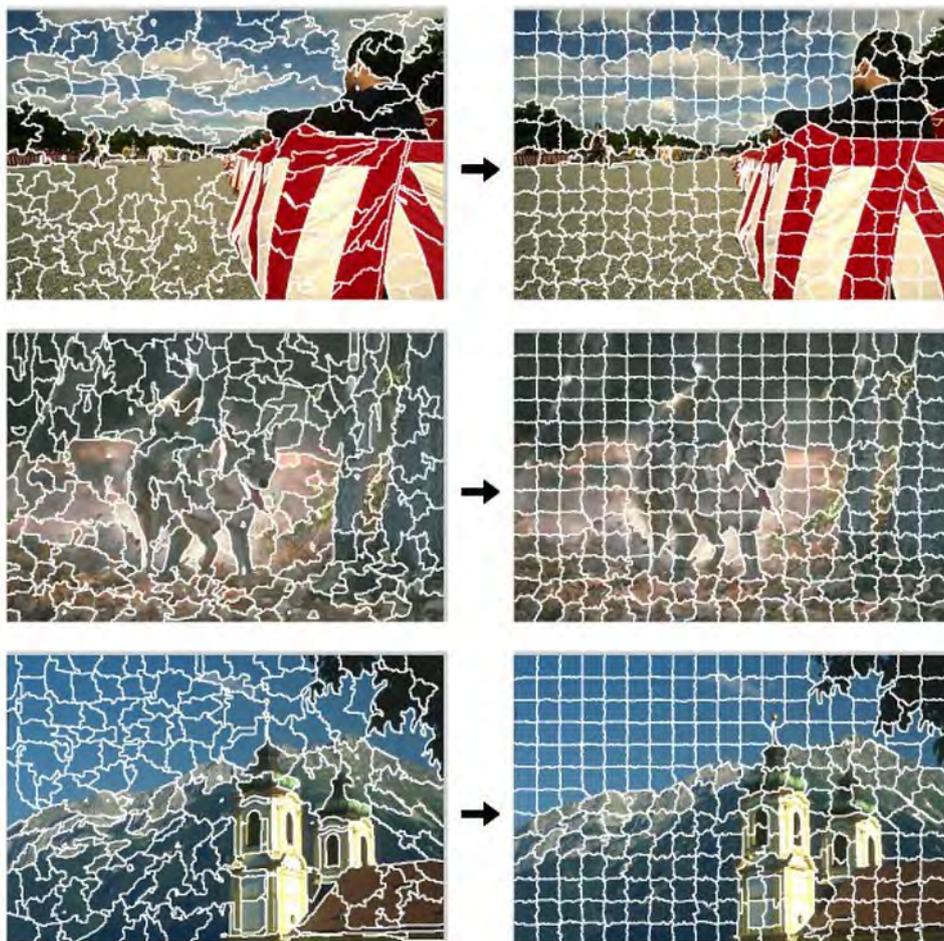
Inconvénients : nécessite une donnée supplémentaire en entrée pour fonctionner correctement (un MNT).

Compact Watershed

#region growing #gradient-ascent-based

Référence : Neubert, Peer, et Peter Protzel. « Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms », 996-1001, 2014. <https://doi.org/10.1109/ICPR.2014.181>.

Principe : il s'agit d'une version de l'algorithme Watershed qui satisfait les propriétés des superpixels.



Comparaison entre l'algorithme Watershed à gauche et l'algorithme Compact Watershed à droite [Neubert et Protzel, 2014]

¹⁴ <https://baharmon.github.io/watersheds-in-grass>

Fonctionnement : Pour cela, il effectue une montée de gradient à partir de marqueurs initiaux qui ont chacun une étiquette différente. Ces marqueurs sont indiqués manuellement ou automatiquement (les minima locaux du gradient de l'image par exemple).

On attribue aux pixels un niveau de priorité qui correspond à l'amplitude du gradient du pixel. La croissance des régions initiales se fait par incorporation des pixels voisins les plus similaires (niveau de priorité/gradient le plus élevé) jusqu'à ce que deux régions se rejoignent. S'il n'y a pas de gradient, les pixels sur un plateau sont répartis entre les marqueurs initiaux des côtés opposés. Le gradient permet d'assurer une homogénéité des régions en niveau de gris, qui représentent l'altitude.

Paramètres implémentation Scikit-image :

- **markers** : nombre de marqueurs initiaux désirés, ou un tableau de valeurs à utiliser. Si ce paramètre n'est pas renseigné, l'algorithme utilise par défaut les minima locaux de l'image.
- **compactness** : permet d'utiliser l'adaptation en superpixels de l'algorithme, le Compact Watershed. Une grande valeur donne des bassins de formes plus régulières.

Complexité : linéaire au nombre de pixels dans l'image.

Inconvénients : Le bruit ou des irrégularités locales entraînent des sur-segmentations. Pour éviter ce problème, soit un pré-traitement du bruit de l'image est réalisé, soit les régions sont fusionnées après la segmentation selon un critère de similarité.

D'autres versions de cet algorithme existent qui définissent autrement les marqueurs initiaux (ex : les maxima locaux d'une fonction de distance pour séparer les objets) ainsi que le critère de segmentation qui n'est pas toujours le gradient ou le contraste, mais peut être la distance aux marqueurs.

Mean-shift

#clustering

#gradient-ascent-based

#mode-seeking

Référence : Comaniciu, D., et P. Meer. « Mean Shift: A Robust Approach Toward Feature Space Analysis ». *Undefined*, 2002. [/paper/Mean-Shift%3A-A-Robust-Approach-Toward-Feature-Space-Comaniciu-Meer/74f4ecc3e4e5b91fbb54330b285ed5214afe2001](#).

Principe : Algorithme de regroupement non-paramétrique qui génère des segments de manière itérative en effectuant une ascension de gradient.

Fonctionnement : l'algorithme déplace chaque point de l'espace caractéristique des pixels vers l'isobarycentre (centroïde) de l'ensemble E des points de son voisinage jusqu'à convergence par estimation d'une densité de probabilité. Ainsi, chaque point converge vers le mode le plus proche suivant approximativement le gradient. [[A. Vedaldi et S. Soatto, 2008](#)].

Avantages : L'avantage de cet algorithme est qu'il n'y a pas besoin de définir le nombre de segments à l'avance. Il peut également s'appliquer sur un espace de points quelle qu'en soit la dimension, ce qui est intéressant si l'on veut prendre en compte plusieurs bandes d'un raster avec des indices spectraux.

Inconvénients : pas moyen de contrôler le nombre, la taille ou la compacité des superpixels directement ([R. Achanta et al., 2010](#)).

LargeScaleMeanShift d'Orfeo Toolbox (QGIS / Monteverdi)

Référence : Michel, Julien, David Youssefi, et Manuel Grizonnet. « Stable Mean-Shift Algorithm and Its Application to the Segmentation of Arbitrarily Large Remote Sensing Images ». *Geoscience and Remote Sensing, IEEE Transactions on* 53 (1 février 2015): 952-64. <https://doi.org/10.1109/TGRS.2014.2330857>.

Principe : Il s'agit d'une version un peu plus rapide de l'algorithme meanshift qui traite l'image en entrée par morceaux et fournit un résultat indépendant du tuilage. Chaque tuile est lissée puis segmentée avec l'algorithme meanshift. Ce dernier regroupe les pixels voisins dont la distance spatiale est inférieure aux paramètres spatialr et ranger, définis par l'utilisateur. Les petits segments sont fusionnés avec la région voisine la plus proche en terme de radiométrie et l'image finale est vectorisée. Chaque région en sortie est décrite avec la moyenne et la variance radiométrique de chaque bande en entrée.

Paramètres :

- **minsize** : les petites régions dont la taille est inférieure à ce paramètre sont fusionnées à une région voisine similaire
- **spatialr** et **ranger** : paramètre spatial et rayon de portée qui servent à moduler la sensibilité de l'algorithme selon la dynamique et la résolution de l'image

Avantages : ceux du Mean-Shift. De plus, les images intermédiaires du traitement sont écrites sur le disque pour limiter la consommation de mémoire.

Inconvénients : temps de calcul longs. Beaucoup de paramètres (seuls les principaux sont présentés) à tester pour trouver la bonne configuration.

Multi-résolution segmentation Baatz (eCognition)

#region growing #local optimisation

Référence : Baatz, M., et A. Schäpe. « Multiresolution Segmentation : An Optimization Approach for High Quality Multi-Scale Image Segmentation ». *J. Photogramm. Remote Sens.*, 2000, 12-23.

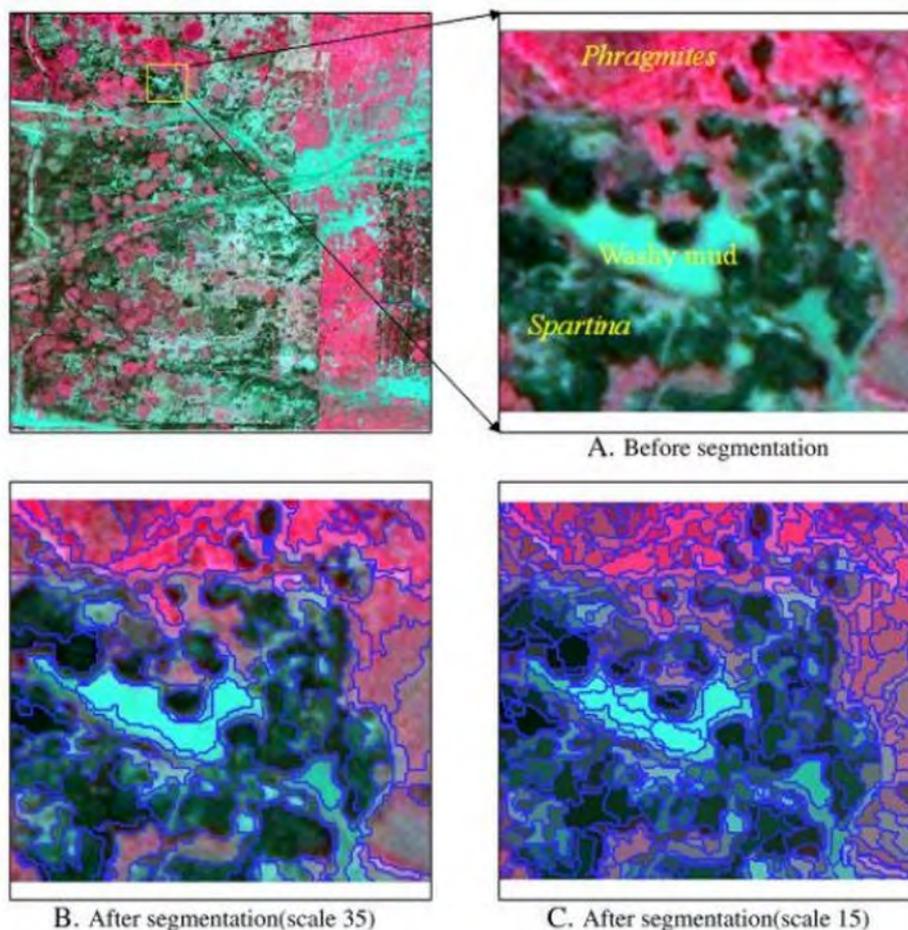


Illustration tirée de Zutao Ouyang, Object-Based Classification & eCognition, 2015

Principe : algorithme de fusion ascendante de régions qui effectue un regroupement par paire. Un processus d'optimisation locale veille à minimiser l'hétérogénéité des segments regroupés.

Fonctionnement : Chaque pixel est considéré comme un objet et l'algorithme va regrouper les objets voisins semblables tout en veillant à minimiser le critère d'hétérogénéité global. Ainsi, à chaque étape, l'algorithme calcule le coût de fusion de paires d'objets adjacents, pondéré par la taille des objets ou la compacité. Seule la paire d'objets au coût le plus faible est fusionnée si ce coût est inférieur à un seuil défini par le facteur d'échelle. Autrement, si aucune fusion n'est possible, l'algorithme s'arrête.

Paramètres :

- *Critère d'hétérogénéité* :

- De *couleur* (somme des écarts-types des valeurs spectrales de chaque canal, pondéré par le poids attribué à chaque canal). Plus la radiométrie des pixels de l'objet est variée, plus l'indice est grand.
- De *forme compacte* (défini par la longueur de la frontière de l'objet divisée par la racine carrée du nombre de pixels). Plus la frontière contient de pixels de l'objet, plus l'indice est grand
- De *forme lisse* (rapport entre la longueur de la frontière de l'objet et la plus courte longueur de frontière possible donnée par la boîte englobante de l'objet). Plus frontière d'un objet est découpée, plus l'indice est grand

Fusion des indices de forme, puis fusion avec l'indice de couleur

- *Facteur d'échelle* : ce seuil contrôle la fin du processus de segmentation. Son augmentation induit un regroupement d'objets plus hétérogènes et donc l'obtention de segments de plus grande taille.

$$\begin{aligned}
 f &= w \cdot h_{\text{couleur}} + (1 - w) \cdot h_{\text{shape}} \\
 h_{\text{couleur}} &= \sum_{b=1}^{n_{\text{bands}}} w_b \cdot \sigma_b \\
 h_{\text{shape}} &= w_s \cdot h_{\text{cmpct}} + (1 - w_s) \cdot h_{\text{smooth}} \\
 h_{\text{cmpct}} &= \frac{p_l}{\sqrt{n_{\text{pxl}}}} \\
 h_{\text{smooth}} &= \frac{p_l}{p_{\text{bbox}}}
 \end{aligned}$$

[Shepherd et al. 2019]¹

Dans la formule le définissant, n_{bands} est le nombre de bandes de l'image en entrée, w [0,1] est un poids équilibrant entre couleur et forme de l'objet dans les critères d'arrêt, w_s équilibre entre compacité et [...] w_b est le poids de la bande b , σ_b est l'écart-type de l'objet pour la bande b , p_l est le périmètre de l'objet, n_{pxl} est le nombre de pixels de l'objet et p_{bbox} est le périmètre de la boîte englobante de l'objet.

Comparaison des paramètres :

Paramètres de segmentation :

- Échelle : 10
- Homogénéité :

Homogénéité	
Couleur	Forme
0.9	0.1
	Compacte
	Lisse
	0.5
	0.5

Test du paramètre de couleur¹⁵

15 BRGM. « Exploitation du logiciel eCognition pour de la classification supervisée par ordinateur », s. d. infoterre.brgm.fr/rapports/RP-52120-FR.pdf.

A facteur d'échelle équivalent, le critère d'hétérogénéité fondé sur la couleur donne produit plus de segments que le critère forme des objets, aussi bien pour le lissage que le compactage.

Paramètres de segmentation :

- Échelle : 10
- Homogénéité :

Homogénéité	
Couleur	Forme
0.1	0.9
	Compacte
	Lisse
	0.1
	0.9

Homogénéité	
Couleur	Forme
0.1	0.9
	Compacte
	Lisse
	0.9
	0.1

Test du paramètre de forme lisse

Test du paramètre de forme compacte

A facteur d'échelle équivalent, le critère d'hétérogénéité fondé sur le lissage donne des objets plus longs et étroits, composés de moins de pixels que les résultats fondés sur le critère d'hétérogénéité de compactage

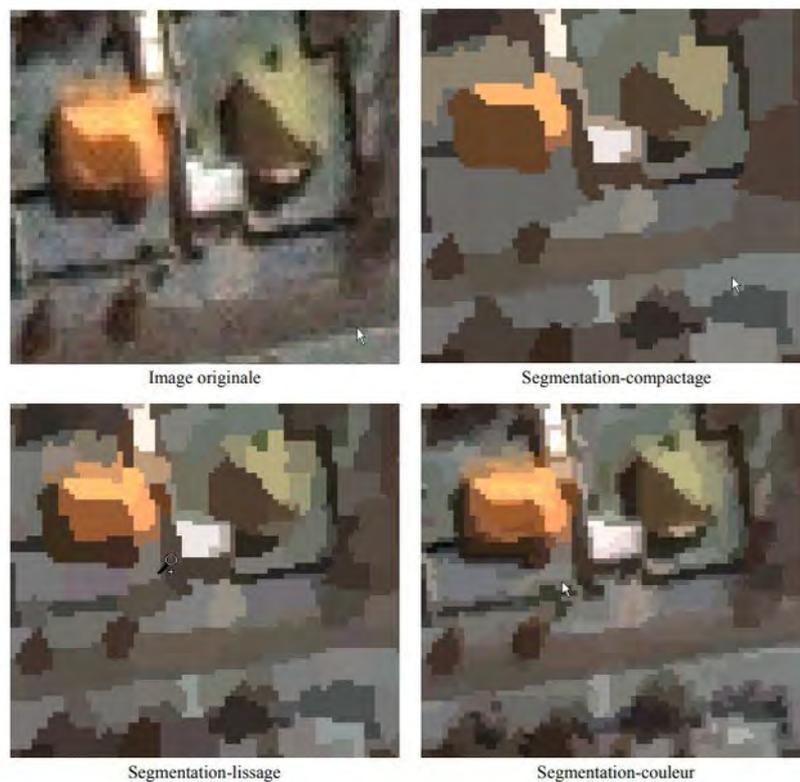


Fig. 2 - Exemple d'objets image générés par les procédés de segmentation avec un choix de paramètre extrême.

Image originale (en haut à gauche), objets image générés issus de la segmentation-lissage (en bas à gauche), objets image générés issus de la segmentation-compactage (en haut à droite), objets image générés issus de la segmentation-couleur (en bas à droite).

Illustration du document du BRGM

Segmentation Multi-échelle Guigues (PYRAM - IGN)

Référence : L. Guigues, R. Trias-Sanz, N. Chehata, F. Taillandier et M. Deveau. « Segmentation multi-échelles d'images : théorie et applications. » 2003.

Principe : Fusion ascendante de régions. Formulation de la segmentation en termes d'énergie. Pour une hiérarchie H de régions donnée, avec $(P_\lambda)_{\lambda \in \mathbb{R}^+}$ l'ensemble des partitions emboîtées, une fonction λ^+ est définie qui calcule pour chaque région r la plus échelle à partir de laquelle r appartient à l'une des partitions de l'image. Celle-ci est appelée par les auteurs *échelle d'apparition*.

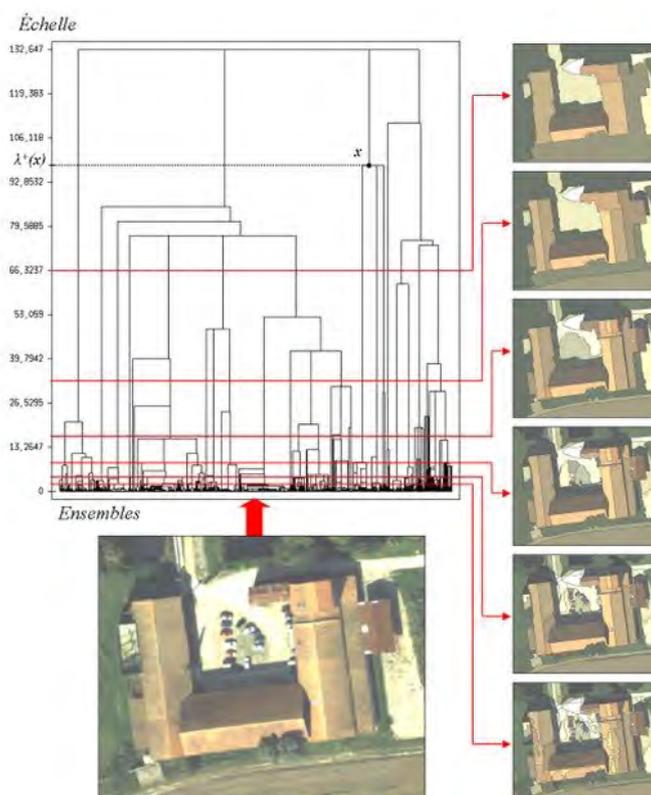


figure B.4.1 : une description ensembles-échelle d'une image et certaines de ses coupes horizontales (à échelle fixe). Le modèle considéré est constant par morceaux, D est la norme L_2 et C est une mesure de complexité privilégiant les formes convexes. La description a été obtenue par l'algorithme d'escalade initialisé à partir d'une sur-segmentation par ligne de partage des eaux de l'image.

[Guigues et al., 2003]

Cette figure illustre le procédé de segmentations emboîtées les unes dans les autres. L'axe des ordonnées représente l'échelle d'apparition de la région.

Fonctionnement : A chaque étape, l'algorithme de programmation dynamique calcule les échelles d'apparition pour chaque couple de régions voisines. L'union qui possède la plus faible échelle d'apparition est fusionnée.

Plus en détails, l'algorithme définit l'énergie totale de la hiérarchie, qui est la somme sur toutes les échelles de l'énergie de ses coupes minimales. En fonction

de l'échelle, l'algorithme modifie les fonctions d'énergie des coupes minimales de la hiérarchie afin d'approcher au mieux le minima global de l'énergie totale.

Autrement dit, les auteurs ont choisi de « procéder comme en généralisation cartographique : partir du niveau de détail le plus fin et progressivement simplifier la description » [Guigues et al., 20003].

Les auteurs ont choisi d'initialiser la segmentation avec un algorithme watershed. Il en résulte N régions, à partir desquels la hiérarchie est constituée. Celle-ci sera composée d'au-plus $2N-1$ régions.

Paramètres : la méthode n'a pas de paramètres mais nécessite d'être initialisée par une segmentation initiale (un algorithme de ligne de partage des eaux calculé sur le gradient de l'image est recommandé) et la définition d'un couple d'énergies.

Avantage : cette méthode est simple, n'ayant pas de paramètres à définir. Elle est invariante aux changements linéaires de la dynamique de l'image. La hiérarchie reste la même, avec un axe des échelles modifié en même proportion. Cet algorithme est intéressant pour les images aériennes, car elle peut s'appliquer à des objets de grande taille, tels un village.

Inconvénient : sur des images aériennes de grande taille, cette méthode se révèle relativement longue à calculer et coûteuse en mémoire.

Segmentation avec un processus d'élimination itérative de Shepherd et al. (RSGISlib)

Référence : Shepherd, James D., Pete Bunting, et John R. Dymond. « Operational Large-Scale Segmentation of Imagery Based on Iterative Elimination ». *Remote Sensing* 11, n° 6 (janvier 2019): 658. <https://doi.org/10.3390/rs11060658>.

Principe : cet algorithme repose sur les k-means et propose un processus d'élimination itératif prenant en compte une unité cartographique minimale.

Fonctionnement : dans un premier temps, chaque bande en entrée est normalisée dans un intervalle de deux écarts-types à la moyenne, avec un seuillage par le minimum et le maximum de l'image. Cela permet d'homogénéiser les dynamiques des bandes. L'image est sous-échantillonnée à 1% ou 10% avant que l'algorithme ne détermine les centres de clusters. Ce sous-échantillonnage accélère grandement les temps de calculs sans que les centres de clusters obtenus ne soient très différents de ceux dérivés de l'ensemble des données d'entrée. L'algorithme calcule alors la distance euclidienne entre les centres de clusters et les caractéristiques de l'image dans l'espace des caractéristiques.

Dans un deuxième temps, il regroupe les pixels pour créer une sur-segmentation de régions de labels uniques.

A la fin de chaque itération, un processus élimine les petites régions qui sont inférieures à l'unité cartographique minimale en les fusionnant au voisin le plus proche d'un point de vue spectral dont la distance spectrale doit être inférieure à 0,1 pour éviter une fusion avec une région de réflectance trop différente.

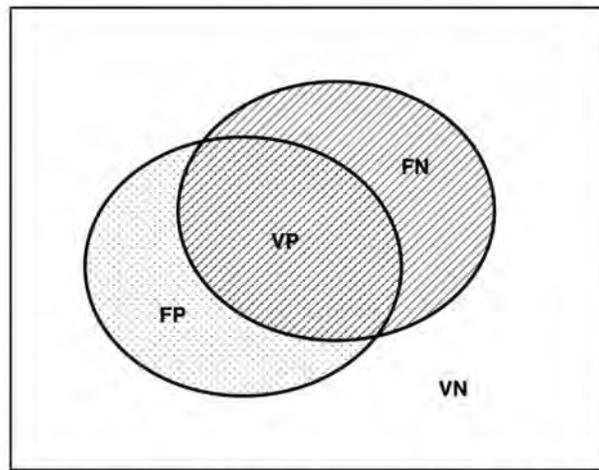
Paramètres :

- Le nombre de centres de clusters initiaux
- L'unité cartographique minimale

Complexité : linéaire, $O(nc)$ avec n le nombre de pixels et c le nombre de centres de clusters.

Avantages : cet algorithme présente de bonnes performances (temps de calcul et gestion de la mémoire). Il peut être appliqué sur de grandes images sans nécessiter de créer une mosaïque de l'image et ne comprend que deux paramètres ce qui permet de l'adapter facilement aux données de précision et d'emprise spatiales diverses.

Critères d'évaluation



VP : Vrai Positif FN : Faux Négatif  Vérité terrain (VT)
 VN : Vrai Négatif FP : Faux Positif  Segmentation (Seg)

$$\text{sensibilité} = \frac{VP}{VP + FN}$$

$$\text{spécificité} = \frac{VN}{VN + FP}$$

$$\text{Jaccard} = \frac{VP}{VP + FP + FN}$$

$$\text{Dice} = \frac{2 VP}{2 VP + FP + FN}$$

Illustration tirée d'un cours de segmentation de l'école Télécom Physique Strasbourg

Bibliographie

Achanta, Radhakrishna, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, et Sabine Süsstrunk. « SLIC superpixels ». *Technical report, EPFL*, 19 juin 2010.

Achanta, Radhakrishna, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, et Sabine Süsstrunk. « SLIC Superpixels Compared to State-of-the-Art Superpixel Methods ». *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, n° 11 (novembre 2012): 2274-82. <https://doi.org/10.1109/TPAMI.2012.120>.

BRGM. « Exploitation du logiciel eCognition pour de la classification supervisée par ordinateur », s. d. infoterre.brgm.fr/rapports/RP-52120-FR.pdf.

Felzenszwalb, Pedro F., et Daniel P. Huttenlocher. « Efficient Graph-Based Image Segmentation ». *International Journal of Computer Vision* 59, n° 2 (septembre 2004): 167-81. <https://doi.org/10.1023/B:VISI.0000022288.19776.77>.

Karakış, S, Aycan Marangoz, et G. Buyuksalih. « ANALYSIS OF SEGMENTATION PARAMETERS IN ECOGNITION SOFTWARE USING HIGH RESOLUTION QUICKBIRD MS IMAGERY », 2006.

Neubert, Peer, et Peter Protzel. « Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms », 996-1001, 2014. <https://doi.org/10.1109/ICPR.2014.181>.

Ren et Malik. « Learning a classification model for segmentation ». In *Proceedings Ninth IEEE International Conference on Computer Vision*, 10-17 vol.1, 2003. <https://doi.org/10.1109/ICCV.2003.1238308>.

Shepherd, James D., Pete Bunting, et John R. Dymond. « Operational Large-Scale Segmentation of Imagery Based on Iterative Elimination ». *Remote Sensing* 11, n° 6 (janvier 2019): 658. <https://doi.org/10.3390/rs11060658>.

Andrea Vedaldi et Stefano Soatto. « Quick Shift and Kernel Methods for Mode Seeking ». In *Computer Vision - ECCV 2008*, édité par David Forsyth, Philip Torr, et Andrew Zisserman, 705-18. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008. https://doi.org/10.1007/978-3-540-88693-8_52.

[Benchmarking of Remote Sensing Segmentation Methods | IEEE Journals & Magazine | IEEE Xplore](#)] Benchmarking of Remote Sensing Segmentation Methods,

Stanislav Mikeš, Michal Haindl, Senior Member, IEEE, Giuseppe Scarpa and Raffaele Gaetano, 2015