Utilisation des images satellites pour améliorer le repérage des logements à Mayotte

Maëlys Bernard
Insee
maelys.bernard@insee.fr

Raya Berova
Insee
raya.berova@insee.fr

Thomas Faria
Insee
thomas.faria@insee.fr

2025-07-30

Abstract Malgré leur disponibilité de plus en plus accrue, les images satellites très haute résolution sont jusqu'ici très peu utilisées au sein de la statistique publique. La manipulation de ces données non conventionnelles n'appartient pas encore aux pratiques usuelles des statisticiens publics, bien qu'elles offrent des opportunités significatives pour améliorer ou créer de nouveaux indicateurs statistiques. Cet article vise justement à démontrer qu'il est possible d'intégrer ces données satellitaires dans les processus statistiques afin d'assurer leur qualité. Mayotte constitue un cas emblématique où les difficultés d'accès au terrain et la forte dynamique de développement urbain complexifient l'organisation des enquêtes cartographiques qui permettent de repérer les logements en amont du recensement. Afin d'orienter les travaux de contrôle de la qualité de ces enquêtes, nous avons exploré l'utilisation des images satellitaires très haute résolution Pléiades (0,5 m de résolution spatiale), en combinaison avec des méthodes d'apprentissage profond, spécifiquement via des modèles de segmentation sémantique. La méthodologie développée consiste en l'entraînement d'un modèle SegFormer, fondé sur une architecture de type Transformer, largement reconnue dans la littérature puisqu'elle est à la base des modèles GPT. Le modèle a été entraîné grâce aux annotations issues du projet CoSIA de l'IGN, qui fournit une référence précise de la couverture du sol. À partir des prédictions du modèle, il est possible de produire des estimations précises des surfaces bâties sur un territoire donné, permettant ainsi d'analyser efficacement leur évolution dans le temps. Les résultats obtenus démontrent une performance élevée dans la détection automatisée des évolutions urbaines dans les DROM. Ces résultats sont disponibles sous la forme d'un tableau de bord interactif permettant aux équipes opérationnelles de cibler efficacement les zones prioritaires pour la réalisation d'enquêtes cartographiques de contrôle sur le terrain.

Après avoir décrit la méthode de prédiction de surface du bâti sur les images satellites, cet article présentera un cas pratique d'utilisation de ces prédictions à travers le contrôle

qualité des enquêtes cartographiques de la population à Mayotte. Depuis 2021, le recensement à Mayotte fonctionne sur un cycle quinquennal comme en métropole et dans les autres DOM. En amont de la collecte auprès des habitants, une enquête cartographique permet de repérer les logements. Le résultat de l'enquête cartographique détermine la base d'immeubles au sein de laquelle est sélectionné l'échantillon1 de l'enquête annuelle de recensement. Ainsi, si l'enquête cartographique ne couvre pas bien l'intégralité du bâti du territoire à enquêter, les populations et nombre de logements déduits de l'enquête annuelle de recensement seront sous-estimés du fait de ce défaut de couverture. Les données satellites permettent de contrôler la qualité des enquêtes cartographiques en comparant l'évolution de la couverture de bâti avec celle du nombre de logements observé suite aux enquêtes cartographiques au sein de chaque îlot2 à enquêter une année donnée. Les îlots sont répartis en cinq groupes de rotation correspondant à l'année à laquelle a lieu l'enquête annuelle de recensement. Afin d'assurer la qualité des enquêtes cartographiques, un deuxième passage sur le terrain a été planifié dans les grandes communes mahoraises pour certains îlots identifiés comme les plus prioritaires. Cet ordre de priorité a été défini par une méthode mobilisant les images satellites pour repérer les plus susceptibles d'être concernés par des manques potentiels dans les enquêtes cartographiques. L'enquête cartographique corrective ciblant ces îlots a permis d'assurer une meilleure couverture de ces enquêtes et d'éviter un risque de sous-estimation des logements.

Table des matières

1 Introduction

Reprendre l'abstract soumis pour faire l'intro Reprendre les slides du séminaire à la DIRAG pour le contexte « Momentum de Varsovie » Repartir de l'origine du projet (demande DROM Mayotte/ Guyane) Parler de l'IGN et de leur travaux avec leur dataset FLAIR et leur modèle CoSIA

2 Méthodologie

Nous allons dans cette partie revenir sur les concepts d'apprentissage profond (*deep learning*) afin d'expliquer notre démarche et l'application de ces méthodes à nos problématiques précises. Nous définirons ensuite les données que nous avons utilisée pour appliquer notre méthodologie.

2.1 Les modèles de d'apprentissage profond pour l'analyse d'images

Dans le domaine de l'apprentissage statistique, les réseaux de neurones profonds en sont un sous-domaine. Si les premiers algorithmes de réseaux de neurones ont été développé dès les années, les réseaux de neurones profonds ont eux fait leur révolution dans les années 2010 grâce notamment aux développement de nouvelles cartes graphiques (GPU) et d'algorithmes optimisées pour celles ci. En effet, leur développement a longtemps été freiné par la quantité de données nécessaires pour obtenir de bons résultats impliquant donc des temps de calculs très longs et des coûts d'annotations prohibitifs. La conception des réseaux de neurones profond est inspirée du fonctionnement du cerveau humain. En effet, l'idée est de représenter les neurones par des fonctions mathématiques très simples qui s'activent si le signal reçu en entrée est suffisamment fort et transmettent alors leur « activation » dans ce cas. Un grand nombre de neurones correctement organisés permet alors de réaliser des opérations plus complexes, résultantes des différentes activations.

Si l'apprentissage profond est devenu la solution de facto pour l'analyse d'images par ordinateur, et notamment les tâches de segmentation sémantique c'est en parti grâce aux travaux fondateurs de Long et al. (2015) qui ont développé les réseaux entièrement convolutifs, une extension de l'architecture des réseaux de neurones convolutif proposé par LeCun et al. (1989).

2.1.a Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN) tirent leur inspiration du cortex visuel des animaux et se composent de deux parties (voir Figure 1) :

1. Une première partie composée par des couches de convolutions successives qui permet d'extraire des prédicteurs de l'image en entrée du réseau;
2. Une seconde partie permettant de classifier les pixels de l'image en entrée à partir des prédicteurs obtenus dans la première partie.

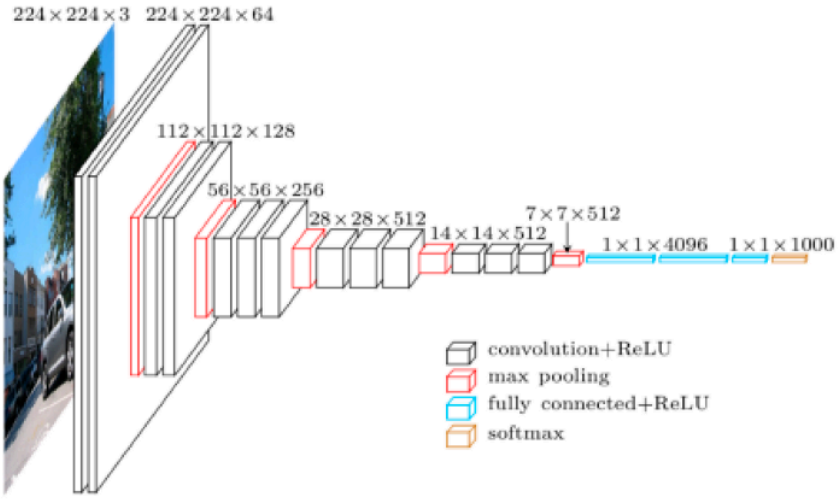

Figure 1. – Réseau de neurones convolutif

Note de lecture : La partie convolutive est représentée par les différents cubes à gauche et la partie classifiante est représentée par les rectangles fins et bleus à droite

Dans la partie convolutive, l'image en entrée du réseau se voit appliquer des filtres appelés convolutions, représentés par des matrices $A = (a_{ij})$ de petite taille appelés noyaux de convolution. L'image en sortie de l'opération de convolution est obtenue à partir de chaque pixel de l'image en entrée en calculant la somme des pixels avoisinant, pondérée par les coefficients a_{ij} du noyau de convolution. Cette opération de convolution est illustrée dans la Figure 2, extraite de l'ouvrage Kim (2017).

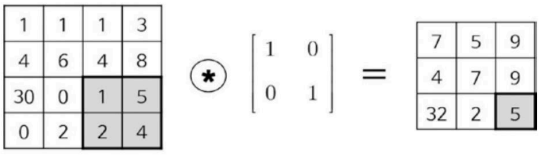

Figure 2. – Opération de convolution

Note de lecture : L'image résultante est une image plus petite dont les pixels sont égaux à la somme des pixels de l'image en entrée pondérée par les coefficients du noyau de convolution.

Une convolution vise ainsi à résumer l'information présente dans une région de l'image. Il est d'ailleurs intéressant de remarquer que lorsque le voisinage d'un pixel ressemble à la structure du noyau de convolution, la valeur de sortie est élevée. Ainsi, si le filtre est, par exemple, un détecteur de ligne verticale, il générera une grande valeur sur les pixels appartenant à des lignes verticales. Un filtre de convolution permet donc d'obtenir une sorte de carte de caractéristiques de l'image. Une couche convolutive correspond finalement à l'application d'un nombre n_f de filtres différents en parallèle. Ainsi, en sortie d'une couche, il y a autant d'images (carte de caractéristiques) que de filtres appliqués. Ces n_f cartes deviennent les canaux d'entrée de la couche suivante.

Les CNN se composent de couches successives, chacune ayant un rôle spécifique. Les premières couches détectent des motifs simples dans l'image (bords, textures, lignes horizontales ou verticales…) tandis que les couches intermédiaires et profondes, plus spécialisées, combinent ces motifs simples pour repérer des formes de plus en plus complexes (motifs, objets, visages…). La Figure 3 schématise un réseau de neurones convolutif.

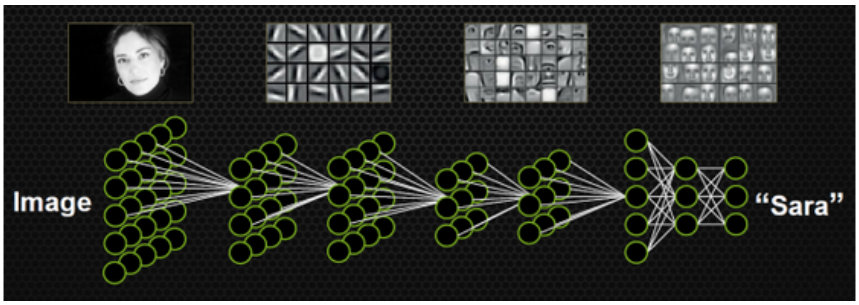

Figure 3. – Représentation d'une convolution

Note de lecture : Les premières couches convolutives reconnaissent les formes simples tandis que les couches les plus profondes à droite reconnaissent des formes plus concrètes.

Après une couche de convolution, la taille des cartes de caractéristiques reste identique à celle de l'image, ce qui peut rapidement entraîner un volume de données important et rendre le modèle coûteux à entraîner. Pour pallier cela, une méthode couramment utilisée est le *max pooling*. Cette opération, représenté par la Figure 4 consiste à diviser chaque carte de caractéristiques en petites régions, puis à ne conserver, pour chacune, que la valeur maximale. Le *max pooling* permet ainsi de réduire la dimension spatiale des données tout en conservant l'information la plus pertinente. Cette réduction favorise également une certaine invariance locale aux petites translations ou déformations de l'image, et contribue à limiter le nombre de paramètres et la complexité de calcul du réseau, tout en mettant en valeur les motifs les plus discriminants. D'autres opérations telles que le padding détaillé dans la Figure 5 permettent malgré tout de contrôler la dimension de l'image en sortie.

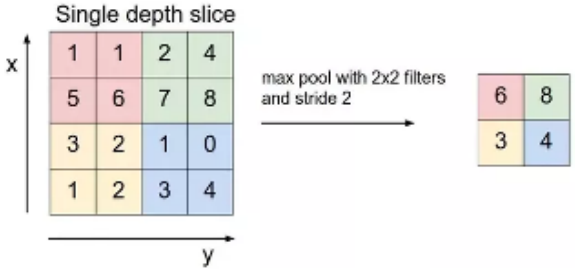

Figure 4. – Représentation du max pooling

Note de lecture : L'image résultante de l'opération de max-pooling est égale au maximum de chaque pixel dans chacune des zones dessinées sur l'image de gauche.

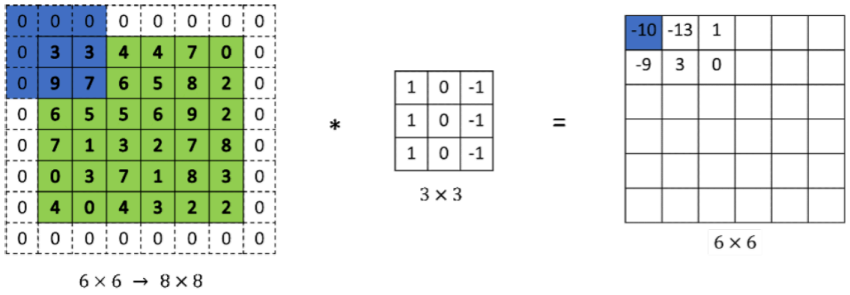

Figure 5. – Représentation du padding

Note de lecture : l'ajout d'une bande de 0 autour de l'image avant d'appliquer la convolution permet de limiter la réduction de dimension de l'image résultante.

Une fonction d'activation non linéaire, comme la ReLU (Rectified Linear Unit), est appliquée systématiquement après chaque couche de convolution (cf. Figure 6). Inspirée du fonctionnement des neurones biologiques, cette opération introduit une non-linéarité dans le réseau. Sans cette étape, l'empilement de couches convolutives ne ferait qu'appliquer une combinaison linéaire de filtres, ce qui limiterait la capacité du modèle à apprendre des relations complexes. L'ajout de fonctions d'activation non linéaires permet ainsi au réseau d'extraire et de modéliser des motifs variés et des structures non linéaires présentes dans les données, ce qui est essentiel pour traiter efficacement des images. L'utilisation de fonctions d'activation non linéaires ne se limite pas aux réseaux convolutifs, c'est justement une composante fondamentale de l'ensemble des architectures d'apprentissage profond.

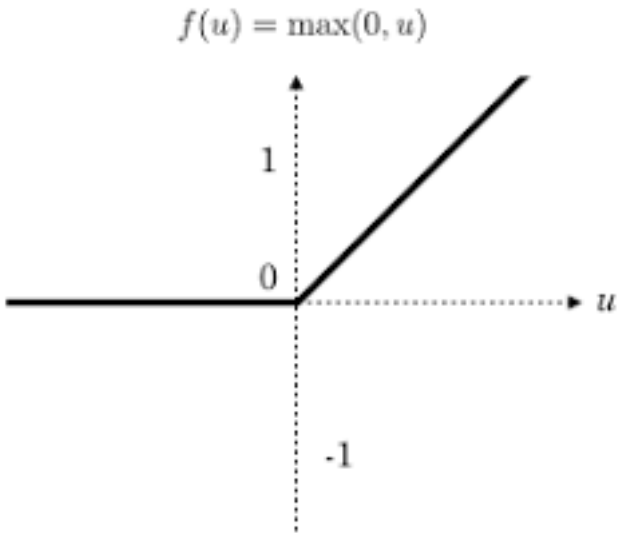

Figure 6. – Fonction d'activation ReLu

Note de lecture : la fonction ReLU « s'active » quand l'argument est positif.

Contrairement aux approches classiques du machine learning, la particularité des réseaux convolutifs réside dans l'automatisation de l'extraction des caractéristiques (*feature extraction*) de l'i-

mage. Dans un cadre traditionnel, cette étape est généralement réalisée de manière manuelle ou repose sur l'expertise métier : on choisit alors à l'avance quels prédicteurs utiliser (par exemple, des statistiques sur certaines bandes spectrales ou des filtres définis à la main). À l'inverse, dans un réseau de neurones convolutif, ce sont les poids des noyaux de convolution eux-mêmes qui sont appris automatiquement au cours de l'entraînement. Ces coefficients font partie du vecteur de paramètres du modèle, noté θ. Ainsi, une fois le réseau entraîné, l'ensemble des filtres optimaux θ^* représente la meilleure façon de transformer et d'extraire les informations pertinentes de l'image. Les cartes de caractéristiques produites par l'enchaînement de ces filtres alimentent ensuite la partie classifiante du réseau. Cette approche permet au modèle de découvrir de manière autonome les motifs les plus discriminants pour la tâche visée, sans intervention manuelle dans la conception des prédicteurs.

En sortie d'un réseau de neurones convolutif classique, la partie dite classifiante permet d'associer une catégorie à l'image analysée. Par exemple, on peut vouloir déterminer si une image représente un chien ou un chat. Pour cela, la sortie de la partie convolutive (c'est-à-dire les cartes de caractéristiques extraites par les filtres) est d'abord transformée en un seul vecteur grâce à une opération de mise à plat (*flattening*). Ce vecteur est ensuite traité par un réseau de neurones dense, appelé aussi *fully connected*, qui produit en sortie un vecteur de scores $x = (x_0, ..., x_9)$ lorsqu'on cherche à classer l'image parmi 10 classes possibles.

Pour interpréter ces scores comme des probabilités, on applique la fonction *softmax* :

$$\text{Softmax } (x_i) = \frac{\exp x_i}{\sum_{j=0}^{9} \exp x_j}.$$

Cette opération convertit les scores en une distribution de probabilité, dont la somme vaut 1, et permet de sélectionner la classe associée à la probabilité la plus élevée comme prédiction finale. La fonction *softmax* est particulièrement utile car elle est infiniment dérivable, ce qui garantit la dérivabilité de tout le modèle f_θ par rapport à ses paramètres θ. Cela rend possible l'utilisation de la descente de gradient pour ajuster les poids du réseau au cours de l'apprentissage.

2.1.b Segmentation sémantique

Comme nous l'avons vu, les réseaux de neurones convolutifs (CNN) sont particulièrement bien adaptés aux tâches de classification d'images. Historiquement, ils ont démontré leur efficacité sur des problèmes comme la reconnaissance de chiffres manuscrits (ex. : MNIST[1]), où l'objectif est d'assigner une étiquette à une image dans son ensemble.

Dans notre projet, la problématique est différente : il s'agit d'identifier automatiquement les zones bâties sur des images satellites. Autrement dit, on ne cherche pas à classifier une image globalement, mais à déterminer pour chaque pixel s'il appartient à une catégorie donnée (bâti ou non bâti, par exemple). Cette tâche s'inscrit dans le cadre de la segmentation sémantique, dont l'objectif est de produire une prédiction dense, c'est-à-dire une carte de labels de même dimension spatiale que l'image d'entrée.

[1] https://fr.wikipedia.org/wiki/Base_de_données_MNIST

La segmentation sémantique a connu un tournant majeur avec l'introduction des *Fully Convolutional Networks* (FCN), notamment dans l'article fondateur de Long, Shelhamer, & Darrell (2015). Ce travail a proposé une adaptation des CNN à la segmentation en éliminant la nécessité d'aplatir l'image avant la classification, une étape qui, dans les CNN traditionnels, conduit à la perte de la structure spatiale de l'image (c'est-à-dire les relations entre pixels voisins), pourtant cruciale pour localiser précisément les objets.

Dans les CNN classiques, on empile des couches de convolution qui extraient des caractéristiques de plus en plus abstraites, et l'on réduit progressivement la résolution spatiale à l'aide du *max pooling*. La profondeur de l'image (le nombre de canaux) augmente, mais sa taille spatiale (hauteur × largeur) diminue. Finalement, les données sont aplaties et envoyées dans une ou plusieurs couches entièrement connectées pour produire une prédiction globale. Mais cette architecture n'est pas compatible avec une prédiction localisée pixel à pixel.

L'idée clé des FCN est de supprimer ces couches entièrement connectées et l'étape d'aplatissement, et de les remplacer par des convolutions 1×1. Cette opération permet de transformer chaque vecteur de caractéristiques (par pixel) en un score de classe, tout en conservant la structure spatiale. On obtient alors, en sortie, une carte par classe, où chaque pixel contient une prédiction de probabilité.

Un problème subsiste toutefois : à cause des opérations de *max pooling* successives, la sortie du réseau est plus petite que l'image d'entrée. Or, pour produire une prédiction par pixel à la résolution d'origine, il est nécessaire de restaurer la taille initiale. Pour cela, Long, Shelhamer, & Darrell (2015) proposent l'utilisation de la convolution transposée (souvent appelée déconvolution, bien que ce terme soit techniquement incorrect), qui agit comme une opération inverse de la convolution en augmentant la résolution spatiale. La Figure 7 illustre schématiquement ce mécanisme.

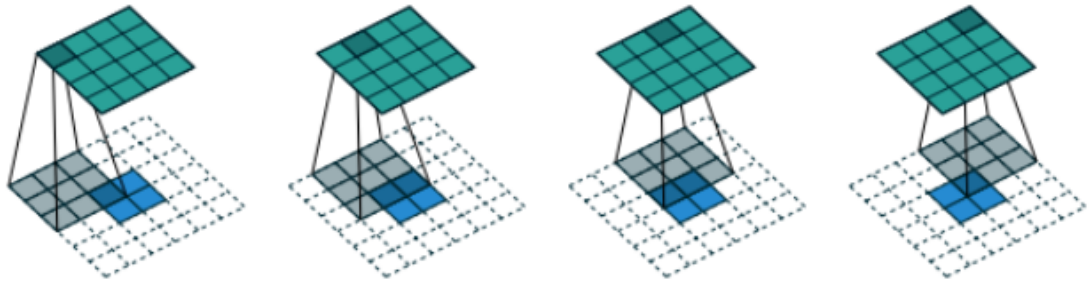

Figure 7. – Représentation de la déconvolution

Note de lecture : TODO.

Pour résumer, la plupart des modèles de segmentation adoptent une architecture en U, composée de deux parties complémentaires :

- La branche descendante (encodeur), qui extrait progressivement des représentations de plus en plus abstraites de l'image. Elle applique des couches convolutives et des opérations de *max pooling* successives afin de réduire la résolution spatiale tout en enrichissant la représentation

sémantique. Le résultat est un embedding (vecteur de caractéristiques) condensé, moins volumineux que l'image initiale. Cette partie est similaire aux CNN.
- La branche montante (décodeur), qui reconstruit une carte de sortie à la même résolution que l'image d'entrée, en utilisant des opérations d'*upsampling* comme la convolution transposée. Cette partie vise à propager l'information sémantique vers les pixels, tout en rétablissant progressivement la structure spatiale.

Il est important de noter que les deux branches sont apprises : les poids des filtres convolutifs et transposés sont ajustés durant l'entraînement, via rétropropagation, comme dans n'importe quel réseau de neurones profond.

Parmi les extensions les plus remarquables aux FCN, on peut citer le modèle U-Net proposé par Ronneberger et al. (2015). Dans les architectures FCN classiques, les couches de convolution couplée au *pooling* induisent une perte progressive de détails spatiaux fins, ce qui tend à produire des contours flous dans les cartes de segmentation. L'idée novatrice des auteurs réside dans sa structure symétrique en U, où chaque niveau de la partie montante est connecté à son équivalent dans la partie descendante via des connexions appelées *skip connections*. Ces connexions permettent au décodeur de récupérer directement les cartes de caractéristiques locales extraites aux étapes de l'encodeur, riches en détails de bas niveau (textures, contours), tout en conservant le contexte global appris dans les couches profondes. La Figure 8 illustre cette mécanique de fusion multi-niveaux.

Une autre avancée majeure est le modèle DeepLabv3, introduit par Chen & al. (2017). Ce dernier cherche également à atténuer la perte d'information due à l'opération de *pooling*, mais en adoptant une approche différente : l'utilisation de convolutions dilatées (*atrous convolutions*). Le principe est d'étendre la taille du noyau de convolution en insérant des espaces (zéros) entre les éléments du filtre. Cela permet d'augmenter le champ réceptif (vision globale du réseau) sans augmenter le nombre de paramètres à estimer.

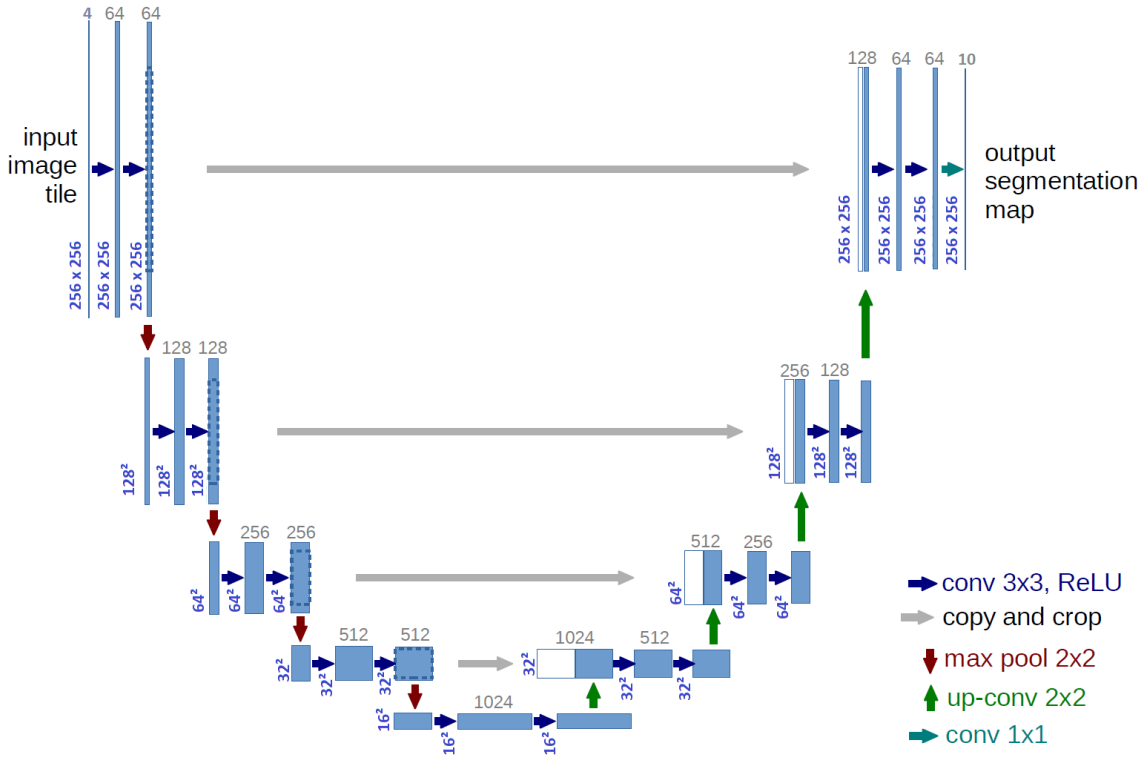

Figure 8. – Représentation schématique du U-Net

Note de lecture : TODO

Malgré les avancées considérables apportées par les architectures convolutionnelles, ces dernières présentent plusieurs limitations structurelles. Par nature, une convolution opère sur un voisinage local (défini par la taille du noyau), ce qui restreint la capacité du réseau à capturer des dépendances à longue distance dans l'image. Pour élargir le champ de vision du modèle, il faut empiler plusieurs couches convolutives, ce qui augmente la profondeur du réseau… et donc son coût computationnel.

Par conséquent, si les CNN sont très efficaces pour modéliser les détails locaux, ils peinent à capturer des relations spatiales globales. Par exemple, deux parties d'un même objet spatialement éloignées (comme les ailes d'un avion ou les extrémités d'une route) sont rarement mises en relation par un CNN standard. De plus, les réseaux convolutionnels ne sont pas naturellement invariants au positionnement global dans l'image : une voiture située en haut à gauche ou en bas à droite peut être traitée différemment, bien qu'il s'agisse du même objet.

La révolution amorcée par les Transformers dans le traitement du langage naturel (NLP), à travers l'article fondateur de Vaswani et al. (2017), a rapidement essaimé vers d'autres domaines, dont celui de la vision par ordinateur. En 2020, les Vision Transformers (ViT), introduits par Dosovitskiy et al. (2021) chercheurs chez Google, ont bouleversé les approches traditionnelles de la vision artificielle. Leur particularité réside dans l'adoption du même mécanisme fondamental que celui utilisé en NLP : le *self-attention*. Ce mécanisme permet au modèle de se concentrer sur différentes

parties d'une image (comme il le ferait avec des mots dans une phrase) en les pondérant dynamiquement par leurs importances, afin de mieux en comprendre la structure et le contenu.

Contrairement aux CNN qui traitent directement l'image comme une grille de pixels 2D, les Vision Transformers commencent par diviser l'image en petits blocs réguliers, appelés *patches*. Chaque *patch* est ensuite aplatit et encodé en vecteur (*embedding*), exactement comme un mot l'est dans un modèle de langage. Il faut donc s'imaginer qu'une image devient une séquence de vecteurs de la même manière qu'une phrase est une séquence de mots. Cependant, contrairement aux mots dans une phrase, les *patches* d'image n'ont pas d'ordre explicite ou structure syntaxique. Pour préserver la structure spatiale, on ajoute à chaque embedding un encodage positionnel (*positional encoding*), qui fournit au modèle une information sur la position d'origine du *patch* dans l'image.

Finalement, une fois tous les *patches* encodés, ils sont passés dans le mécanisme de self-attention, qui permet à chaque *patch* de s'informer de tous les autres *patches*, en attribuant à chacun une pondération calculée dynamiquement selon leur pertinence. Ce mécanisme permet de capturer à la fois des dépendances locales (entre *patches* voisins) et globales (entre régions éloignées), ce qui est particulièrement utile pour des objets de grande taille ou des structures étendues.

Ce traitement est généralement répété à travers plusieurs couches de Transformer (notées « $L \times$ » dans la Figure 9), permettant un enrichissement progressif des représentations à chaque niveau. À la sortie, chaque *patch* est représenté par un vecteur qui incorpore à la fois son propre contenu et son contexte global.

On obtient alors de nouveaux vecteurs qui sont des représentations des *patches* individuels, enrichies par le contexte global. Il est important de noter qu'on empile généralement plusieurs couches Transformer (par exemple, « $L \times$ » signifie L couches dans la Figure 9). Dans la configuration originale du ViT pour la classification d'image, un token spécial est ajouté à la séquence dès l'entrée. Après passage dans les couches Transformer, le vecteur final associé à ce token est utilisé comme représentation globale de l'image, que l'on transmet à une couche entièrement connectée pour effectuer la classification.

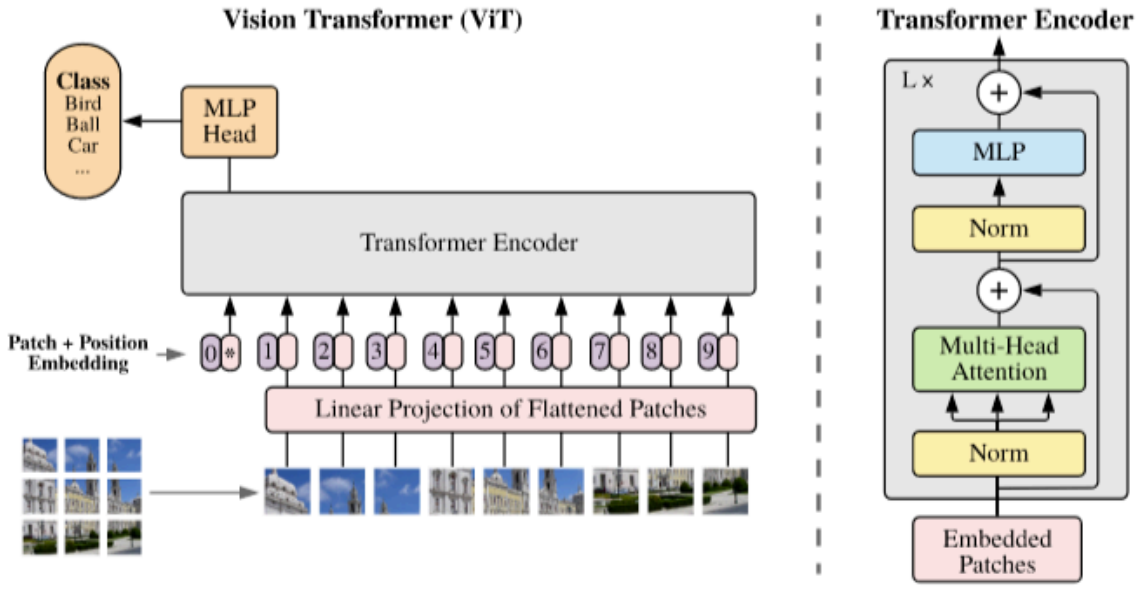

Figure 9. – Représentation schématique du Vision Transformer (ViT)

Note de lecture : TODO

Le succès des Vision Transformers (ViT), initialement conçus pour des tâches de classification d'images, a rapidement suscité un intérêt croissant pour leur adaptation à des tâches plus complexes, notamment la segmentation sémantique. Cependant, l'utilisation directe des ViT pour la segmentation se heurte à plusieurs limitations :

1. Les ViT standard ne conservent pas la structure spatiale de manière aussi précise que les CNN, et l'encodage positionnel appris se révèle souvent insuffisant pour localiser finement les objets au niveau du pixel les ViT standard ne conservent pas la structure spatiale de manière aussi précise que les CNN, et l'encodage positionnel appris se révèle souvent insuffisant pour localiser finement les objets au niveau du pixel.
2. Contrairement aux CNN qui construisent des représentations à différentes échelles (petits détails et vue globale), le ViT standard traite tous les *patches* à la même résolution.
3. Le mécanisme d'attention a un coût quadratique avec la taille des *patches*. Plus on veut de détails (donc des *patches* plus petits), plus ça devient lourd à calculer.

Pour répondre à ces défis, le modèle SegFormer, proposé par Xie et al. (2021) chez NVIDIA, introduit une approche hybride qui combine intelligemment les forces des CNN (efficacité locale et structure hiérarchique) avec celles des Transformers (apprentissage du contexte global via self-attention).Le modèle SegFormer, est défini par deux composantes principales (cf. Figure 10) un encodeur (hiérarchique) basé sur des couches Transformer empilées inspiré des CNN et un décodeur très simple pour produire la carte de segmentation.

Contrairement au ViT standard qui applique le même traitement à tous les patches, SegFormer adopte une structure à plusieurs niveaux de résolution, analogue à celle des CNN. L'image est

traitée par une succession de blocs Transformer, où chaque niveau réduit progressivement la résolution spatiale, construisant ainsi une hiérarchie d'abstractions.

De plus, le partitionnement de l'image se fait en *patches* chevauchants (*Overlapping Patch Merging*). Contrairement aux ViT où les *patches* sont non recouvrants (et donc spatialement disjoints), ici chaque *patch* inclut une portion de ses voisins, ce qui permet de préserver la continuité spatiale. Grâce à cette conception, et à la structure hiérarchique à plusieurs résolutions, il n'est plus nécessaire d'utiliser un encodage positionnel explicite : la position est captée implicitement par le recouvrement et la profondeur du traitement.

Finalement, l'une des particularité du Segformer est également son décodeur très simple qui contraste avec les décodeurs complexes de l'U-Net ou DeepLab. En effet, il prend les sorties des différents niveaux - 4 dans le papier - de l'encodeur qui représentent des résolutions différentes. Il les projette dans un espace commun, les interpole à la même taille, puis les concatène pour produire la carte de segmentation via quelques couches simples. Cette conception permet au SegFormer de rester relativement léger en nombre de paramètres, tout en offrant des performances compétitives. En conséquence, le modèle est rapide à entraîner, efficace à l'inférence, et bien adapté aux contraintes de déploiement en production.

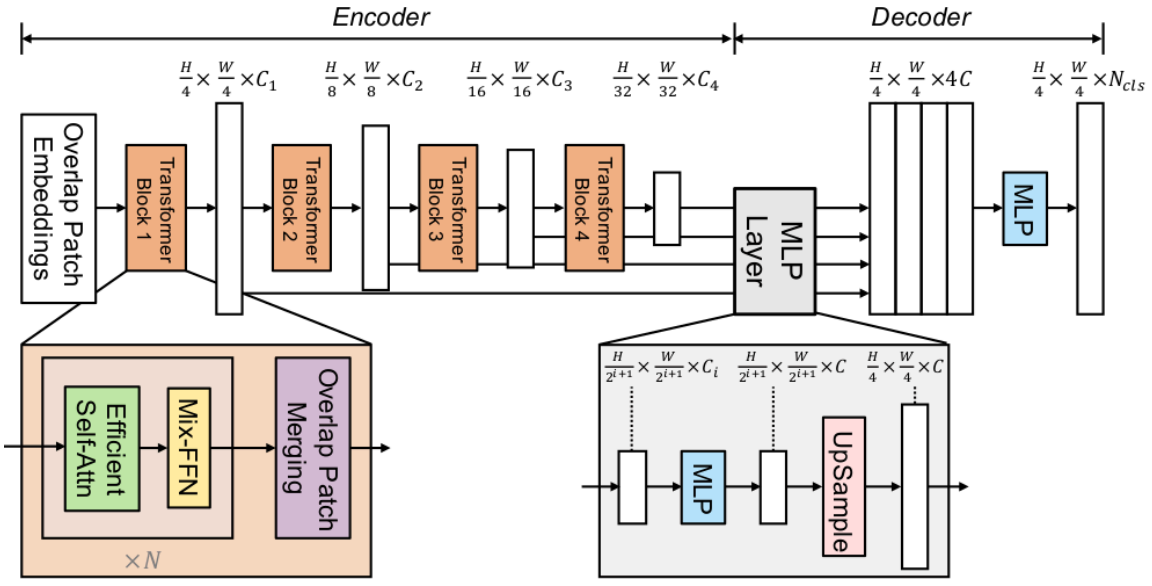

Figure 10. – Représentation schématique du Segformer

Note de lecture : TODO

Parmi les modèles de segmentation récents considérés comme étant à l'état de l'art, on peut citer SETR, proposé par Zheng et al. (2021), qui fut l'un des premiers à adapter une architecture Transformer pure à la segmentation sémantique. Le Swin Transformer, introduit par Liu et al. (2021), repose quant à lui sur une approche hiérarchique utilisant des fenêtres glissantes à déplacement progressif (*shifted windows*), permettant de mieux capter les structures à différentes échelles tout en conservant une efficacité computationnelle élevée. Enfin, le Segment Anything Model v2 (SAM-2), récemment développé par Meta (Ravi et al. (2024)), représente une avancée

majeure dans la segmentation universelle. Il permet de segmenter automatiquement n'importe quel objet dans une image, à partir d'un simple point d'indication ou d'un prompt, sans entraînement spécifique à un domaine donné.

Dans le cadre de notre projet, après avoir évalué plusieurs architectures, notamment DeepLabv3, notre choix s'est porté sur le modèle SegFormer-B5[2], avec lequel nous avons obtenu les meilleurs résultats en termes de précision de segmentation, tout en maintenant des temps de calcul raisonnables. Nous avons utilisé la version pré-entraînée du modèle proposée par NVIDIA, que nous avons ensuite spécialisée via un apprentissage supervisé sur nos images satellites des DROM. Ce transfert d'apprentissage nous a permis de bénéficier à la fois des connaissances générales acquises sur de grands jeux de données, et d'une adaptation fine aux caractéristiques particulières de la géographie des DROM.

2.2 Données

2.2.a Images satellites

Une image satellite est une matrice à trois dimensions. Chaque élément de cette image est un **pixel**, qui correspond à une surface au sol (par exemple 10m*10m). Le pixel contient plusieurs valeurs numériques. Ces valeurs expriment l'intensité du rayonnement solaire reflété dans chaque **bande spectrale** pour ce pixel. Une bande spectrale correspond à une portion du spectre électromagnétique, qui peut être par exemple le bleu, le rouge, le vert, le proche infrarouge. Donc, pour une image satellite de 200*200 pixels avec les trois bandes du visible (rouge, vert, bleu), chaque pixel aura trois valeurs différentes représentant l'intensité dans chacune des bandes du visible. Ainsi, un pixel qui représente 10m*10m au sol donnera une couleur : du violet sur l'exemple de la Figure 11.

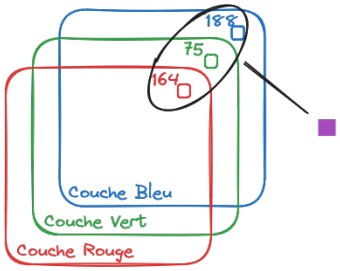

Figure 11. – Exemple d'un pixel d'une image satellite

Il existe de nombreux produits satellitaires disponibles. Nous nous sommes surtout concentrés sur deux d'entre eux : **Pléiades et Sentinel2**.

[2]https://github.com/NVlabs/SegFormer

Figure 12. – Mamoudzou, Mayotte (2024)

Les images satellites **Pléiades** constituent une ressource précieuse dans notre cas d'usage. Ce sont des images à très haute résolution, spécifiquement conçues pour l'observation fine des territoires. Elles offrent trois bandes spectrales dans le visible (**rouge, vert, bleu**) auxquelles s'ajoute une quatrième bande dans le proche infrarouge (NIR), bien que cette dernière ne soit pas disponible dans nos données actuelles. Leur résolution spatiale remarquable de **0,5m** par pixel permet une détection très précise des objets au sol, ce qui est essentiel pour nos traitements d'analyse fine.

Ces images peuvent être obtenues soit via les **archives gratuites** (sous conditions d'accord), soit par acquisition à la demande, un service payant encadré par une licence Airbus©, avec des délais d'environ 6 à 8 mois par département.

Dans le cadre du cyclone Chido qui a frappé Mayotte en décembre 2024, un plan d'urgence a permis l'accès gratuit à une mosaïque d'images Pléiades post-cyclone couvrant l'île. Une **mosaïque** est une image composite constituée d'assemblages de prises de vues réalisées à différentes dates. Elle permet d'obtenir une couverture complète, avec un minimum de zones nuageuses et une meilleure homogénéité visuelle. Ce travail de reconstitution, mené par **l'Institut National de l'information géographique et forestière** (IGN), est crucial pour garantir des données exploitables, notamment dans le cadre de l'entraînement de modèles d'analyse automatique.

Mais Pléiades n'est pas notre seule source d'imagerie satellite. Les satellites **Sentinel-2**, développés par l'Agence spatiale européenne (ESA) dans le cadre du programme Copernicus, offrent une alternative open source, particulièrement intéressante pour les analyses à large échelle ou à forte fréquence temporelle. Contrairement à Pléiades, Sentinel-2 capte **treize bandes spectrales**, réparties entre le visible, le proche infrarouge (NIR) et l'infrarouge à ondes courtes (SWIR), ce qui

ouvre la voie à une multitude d'indices et d'analyses thématiques (comme la détection de la végétation, de l'humidité, ou des matériaux).

En revanche, la résolution spatiale de Sentinel-2 est moindre : selon les bandes, elle varie entre **10 m**, 20 m et 60 m, ce qui rend l'observation de petits objets ou de détails fins plus difficile. Néanmoins, ces images ont l'avantage d'être acquises automatiquement **tous les cinq jours**, garantissant une fréquence de revisite élevée et une mise à disposition régulière des données, y compris en période de crise.

Ainsi, les images Pléiades semblent être les plus adaptés pour de la détection de bâtiment dans les DROM.

- definition
- produits
- acquisition/livraison (partenariat IGN a promouvoir)

Faire un atbleau récapitulant toutes les données que l'on a evec le millesime

2.2.b Annotations

- Rappeler que c'est le plus important pour avoir de bons modèles
- Nous n'avons rien annoté manuellement => full automatique
- Rappel du coût d'annotation pour en avoir de qualité (en ETP ?)
- RIL (Insee),
- BDTOPO (better)
- CoSIA

peut etre rappeler la difficulté d'avoir des couples qui sont iso temporel

3 Résultats

3.1 Entraînement et évaluation

- IOU, Loss, zone de test manuel (calculer la taille des zones de test)

Metriques et images

3.2 Inférence

Dans le cadre de notre projet de détection automatique des zones bâties à partir d'images satellites, il est essentiel de distinguer les phases d'entraînement du modèle et d'inférence. L'entraînement correspond à la phase exploratoire du projet, mobilisant des ressources matérielles conséquentes (notamment des GPU) et nécessitant de nombreux choix méthodologiques (prétraitement des images, ajustement des hyperparamètres, choix du modèle, etc.). Cette étape, une fois réalisée, n'est plus réellement centrale et peut être amélioré à la marge de manière indépendante.

À l'inverse, la phase d'inférence s'apparente à une mise en production opérationnelle du modèle entraîné. Une fois les poids du modèle estimés, la génération de prédictions devient relativement peu coûteuse, notamment grâce à la possibilité d'exécuter les inférences sur CPU avec un temps de latence acceptable. L'enjeu devient alors de concevoir un dispositif fiable, reproductible et facilement mobilisable pour mettre à disposition les résultats du modèle, en particulier lorsque de nouvelles images satellites sont acquises.

Actuellement, les inférences sont réalisées sur des tuiles disjointes de taille 250×250 pixels, qui correspondent à la taille utilisée lors de l'entraînement. Chaque image satellite de 2000×2000 pixels est ainsi découpée en 64 sous-images traitées indépendamment. Cette stratégie, bien que simple, présente plusieurs inconvénients. Elle introduit des artefacts de bordure, limite la vision contextuelle du modèle, et empêche une prise en compte fluide des objets situés à la jonction de deux tuiles.

Afin d'améliorer la cohérence spatiale des prédictions, nous mettons en œuvre une stratégie d'inférence par fenêtre glissante (*sliding window*). Celle-ci consiste à réaliser plusieurs prédictions pour un même pixel, à partir de fenêtres décalées les unes par rapport aux autres. Les probabilités ainsi obtenues sont ensuite moyennées, ce qui permet de lisser les effets de bord, de renforcer la stabilité des contours et de réduire le bruit.

Cette inférence multiple est complétée par un pipeline de post-traitement multiclasse. Celui-ci repose d'abord sur une stratégie de repli contextuel : pour les pixels dont la probabilité maximale est inférieure à un certain seuil, la classe est réattribuée en fonction du voisinage local (via vote majoritaire ou moyenne pondérée des probabilités). Ensuite, des opérations de morphologie mathématique sont appliquées classe par classe afin de supprimer les artefacts et lisser les masques de segmentation. Enfin, un filtrage par taille minimale est mis en œuvre, avec des seuils spécifiques à chaque classe (par exemple, un bâtiment peut être plus petit qu'une zone d'eau). Ce traitement permet d'éliminer les objets aberrants ou trop petits pour être statistiquement significatifs.

Afin d'industrialiser l'ensemble de la phase d'inférence, une API a été développée. Elle encapsule l'ensemble des étapes précédentes (inférence, post-traitement) et permet une exploitation simple et flexible du modèle par les différentes équipes concernées. Trois points d'entrée sont disponibles :

- fourniture directe d'une image satellite à prédire,
- sélection d'une zone géographique via coordonnées GPS (bounding box) et année d'observation,
- saisie d'un identifiant d'îlot (entité géographique infra-communale) pour obtenir les prédictions correspondantes.

L'API intègre un mécanisme de cache qui évite de recalculer une prédiction déjà effectuée pour une même zone géographique et un même modèle. L'ensemble est déployé sur CPU et permet une chaîne de traitement entièrement automatisée, assurant une inférence rapide dès la réception de nouvelles images satellites.

Conçue comme un outil évolutif au service des équipes métiers de l'Insee, l'API est amenée à s'adapter aux besoins exprimés. Par exemple, à la suite d'un besoin identifié concernant le calcul de statistiques par îlot, une nouvelle fonctionnalité a été ajoutée pour retourner non seulement la carte de prédictions, mais aussi des indicateurs agrégés, tels que les surfaces de bâties par îlot.

Néanmoins, la diffusion de résultats via une API n'est pas toujours la modalité la plus adaptée aux usages internes. C'est pourquoi deux modes de restitution complémentaires ont été développés : d'une part, la production de fichiers Parquet contenant les résultats structurés pour exploitation statistique ; d'autre part, une application interactive cartographique, destinée aux agents de terrain. Cette application s'appuie sur un GeoServer pour diffuser les images satellites, les cartes de segmentation, ainsi que les évolutions détectées (créations, destructions de bâti), facilitant ainsi le croisement avec les informations d'enquête et les validations sur le terrain.

3.3 Mise à disposition pour les statisticiens

3.3.a Vers de nouveaux indicateurs statistiques ?

- Le développement d'un modèle performant n'a de valeur que s'il permet de produire des informations utiles pour la statistique publique.
- L'enjeu est désormais d'identifier quels indicateurs pertinents et exploitables peuvent être dérivés à partir des prédictions du modèle. Partenariat avec des équipes de l'insee
- Des fichiers au format Parquet ont été générés pour tracer l'évolution du bâti dans le temps, constituant une première base exploitable pour des analyses statistiques. Recensement

3.3.b Une application interactive pour les agents de terrain

- Une application web interactive a été développée pour permettre une consultation directe des résultats
- Cette application s'appuie sur un GeoServer qui héberge et diffuse les images satellites et les cartes de segmentation associées.
- Elle offre une interface cartographique permettant aux agents de l'Insee en charge des enquêtes de terrain :
  - d'accéder aux images avant/après sur une zone donnée,

- de visualiser les détections de destructions et de créations de bâtiments,
- et ainsi de mieux préparer les opérations de terrain ou de **vérifier des résultats étranges post collecte.

Dans tous les cas l'enjeu c'est les données. Plus tot on a accès à des données le mieux c'est => partenariat avec l'IGN indispensable

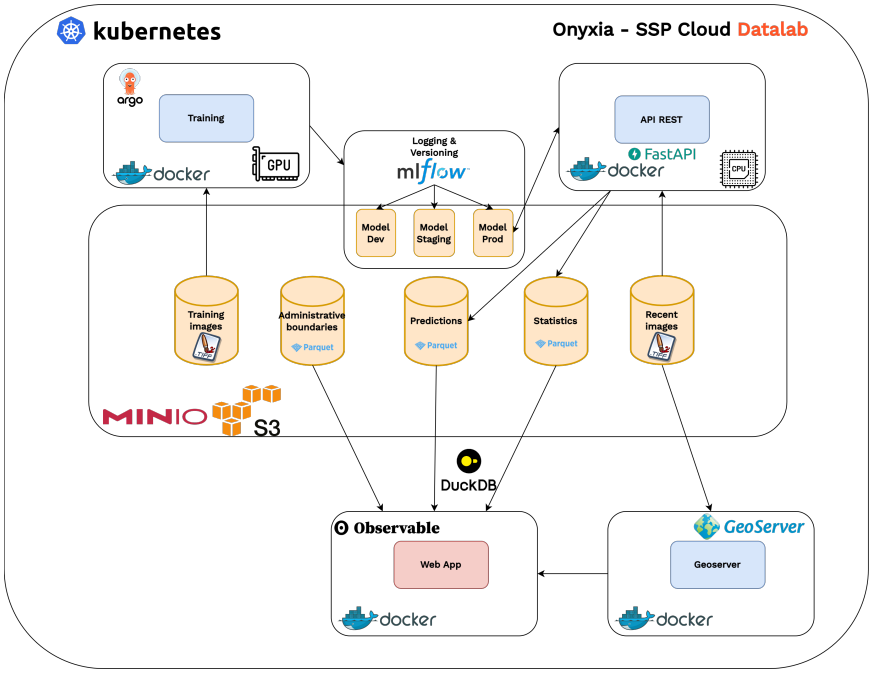

Figure 13. – Schéma de la pipeline

4 Cas d’usage

Bibliographie

Chen, L.-C., & al. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv*. http://arxiv.org/abs/1706.05587

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., & others. (2021). An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*. http://arxiv.org/abs/2010.11929

Kim, P. (2017). Convolutional Neural Network. In P. Kim (éd.), *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence* (p. 121-147). Apress. https://doi.org/10.1007/978-1-4842-2845-6_6

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), 541-551. https://doi.org/10.1162/neco.1989.1.4.541

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. https://arxiv.org/abs/2103.14030

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully Convolutional Networks for Semantic Segmentation*. https://arxiv.org/abs/1411.4038

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). *SAM 2: Segment Anything in Images and Videos*. https://arxiv.org/abs/2408.00714

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*. http://arxiv.org/abs/1505.04597

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Xie, E., Wang, W., Yuille, A. L., Anandkumar, A., & Alvarez, J. M. (2021). SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *arXiv preprint arXiv:2105.15203*. http://arxiv.org/abs/2105.15203

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H. S., & Zhang, L. (2021). *Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers*. https://arxiv.org/abs/2012.15840