Objectifs

L’objectif initial de mon travail était de développer un modèle capable de détecter les éléments de platine présents dans les images microscopiques. Pour atteindre ce but, j’ai opté pour une approche de segmentation d’instance en utilisant le modèle Detectron2.

Base de données

La base de données à ma disposition comprenait un total de 131 images. Pour l'entraînement du modèle, 112 images ont été utilisées afin de permettre au réseau d'apprendre les caractéristiques essentielles et de s'ajuster aux différentes variations présentes dans les données. Les 19 images restantes ont été réservées pour les tests, afin de pouvoir évaluer les performances du modèle et valider sa capacité de généralisation.

L'architecture U-Net

Avant de m'attarder davantage sur le modèle **Detectron2**, j’ai testé le modèle **U-Net** avec les fonctions de perte que Monsieur XXX m’a envoyées, en expérimentant différentes configurations pour les poids weight\_zero et weight\_one. Ces paramètres permettent de moduler l’importance des erreurs des classes négatives (arrière-plan) et positives (objets détectés). Par exemple, dans une première configuration, j’ai choisi weight\_zero = 0.25 et `weight\_one = 1. Cette configuration a donné de meilleurs résultats que ceux présentés lors de ma précédente présentation, notamment pour la détection des petits objets. Cependant, comme illustré dans les images, un problème persiste : le bruit est confondu avec les atomes isolés. Dans les parties noires des images, où aucun objet ne devrait être détecté, plusieurs faux positifs apparaissent.

J’ai alors essayé d’augmenter weight\_zero à 0.5, puis à 0.6, mais le problème persiste. Lorsque j’ai diminué weight\_zero à 0.1, les performances se sont dégradées drastiquement. J’ai également testé une configuration symétrique avec weight\_zero = 0.5 et weight\_one = 0.5, ainsi qu’une configuration avec weight\_zero = 0.5 et weight\_one = 0.8, mais le bruit reste confondu avec les atomes isolés.

En conclusion, bien que l’utilisation de cette fonction de perte améliore la détection des petits éléments, le modèle **U-Net** n’est pas très robuste pour notre cas d’utilisation. Il présente une confusion entre les atomes isolés et le bruit, et en cas de chevauchement, les objets détectés sont regroupés en un seul, ce qui est problématique pour une segmentation précise. Cela confirme que le **U-Net** n’est pas adapté pour ce projet, et que l’approche la mieux indiquée reste la **segmentation d’instances**, notamment avec **Detectron2**.

Detectron2

L'architecture de Detectron2, illustrée ici, est composée de plusieurs modules organisés en deux étapes principales. Tout d'abord, l'image d'entrée passe par un backbone (ResNeXt-101) qui extrait des caractéristiques à différents niveaux de profondeur (C2, C3, C4, C5). Ensuite, ces caractéristiques sont traitées par le module FPN (Feature Pyramid Network), qui produit des cartes de caractéristiques à plusieurs résolutions (P2 à P6) pour capturer des détails à différentes échelles. Les couches P2 et P3 se concentrent sur les objets de petite taille, tandis que P5 et P6 détectent les objets plus grands. Dans la première étape, un RPN (Region Proposal Network) utilise ces cartes de caractéristiques pour générer des propositions de régions d'intérêt (ROI) en produisant une classification binaire (détection d'objet ou arrière-plan) et des ajustements de boîtes englobantes. En Stage 2, les ROIs sont traitées par le module Mask R-CNN qui génère des prédictions pour chaque région proposée en produisant des classes, des boîtes englobantes et des masques pour la segmentation finale. Le réseau de sortie permet ainsi de classifier, localiser et segmenter précisément les objets détectés.

Anchor size

Le paramètre en question détermine les dimensions des ancres pour la détection d'objets, des milliers d’ancres sont généré, représentant l'emplacement, la forme et la taille idéale de l'objet à détecter. Notre objectif étant de détecter des objets plus petits, nous avons modifié ce paramètre, initialement défini à [[32, 64, 128, 256, 512]], pour qu'il soit [[8], [16], [32], [64], [128]]. Avec la configuration par défaut, les objets qui apparaissent dans l'image peuvent ne pas avoir une intersection de plus de 50% avec l'une des boîtes d'ancrage, empêchant ainsi le réseau de neurones de les détecter et de les apprendre.

Results

Cette fois-ci, les métriques de précision et de rappel peuvent être considérées, car je les ai construites moi-même en utilisant un critère strict : si l'intersection entre la prédiction et le ground truth dépasse 50 %, la prédiction est considérée comme un vrai positif. En reprenant les prédictions du modèle **Detectron2** précédemment entraîné et en les utilisant comme nouveaux ground truth pour un nouvel entraînement du modèle, j'ai observé une nette amélioration des performances. Après cette étape, le modèle atteint une précision et un recall d'environ 0.8.

Results

Comme vous pouvez le constater, le modèle détecte un nombre significativement plus élevé d'éléments que le ground truth, en particulier les petits objets tels que les atomes isolés. Le diamètre moyen des nanoparticules (NP), qui est légèrement inférieur à celui indiqué par le ground truth. Cela est dû au fait que, dans le ground truth, certains éléments ne sont pas toujours représentés avec leur taille réelle, comme montré dans la diapo précédente.

Comme vous pouvez le constater dans l'image, la différence de la valeur moyenne des pixels des objets détectés par rapport au fond dans un périmètre voisin, tracée en fonction du diamètre des particules en nanomètres, permet d'identifier clairement les trois classes définies. Les objets dont la valeur est inférieure à 0,2 correspondent aux **atomes isolés**, ceux compris entre 0,2 et 0,5 sont des **clusters**, et ceux dont la valeur dépasse 0,5 sont classifiés comme des **nanoparticules**.

Voici un deuxième exemple appliqué à une autre image. On peut observer que les objets présentant une différence conséquente entre la valeur moyenne des pixels et le fond sont, a priori, situés en **frontière**, c'est-à-dire dans des zones où les voisins se trouvent principalement dans la partie noire de l'image.

Les objets ayant une différence proche de 0 sont difficilement différenciables de leurs voisins et pourraient éventuellement être considérés comme des **faux positifs**, pouvant ainsi être supprimés lors d'un traitement ultérieur. Par ailleurs, les objets présentant une différence négative, généralement autour de 5 par image, semblent inhabituels. Ces objets apparaissent probablement dans des **zones normales**, mais leurs voisins appartiennent à des zones **très blanches**

Pour conclure, le modèle de segmentation d'instance **Detectron2** s'est avéré plus performant que le modèle de segmentation sémantique **U-Net** dans notre contexte d'utilisation et sur nos données. Detectron2 détecte un plus grand nombre d'éléments que la vérité terrain, notamment pour les **atomes isolés**. Étant donné que nous ne disposons pas de vérités terrain spécifiques pour ces cas, une vérification visuelle des résultats sur plusieurs images de test a été effectuée. Cette analyse nous permet de conclure que les inférences du modèle sont **logiques** et **cohérentes** : il ne génère pas de détections arbitraires ("hallucinations"). Cependant, quelques **faux positifs** peuvent apparaître, mais ils peuvent être éliminés en s'appuyant sur la différence de valeurs de pixels entre les objets détectés et le fond.