A dark blue vertical bar on the left side of the slide. A blue arrow points to the right from the bar, containing the date.

24/01/2023

U-Net : Réseaux convolutifs pour le biomédical

Segmentation des images

Several thin, curved lines in dark blue and light grey that sweep upwards from the bottom left corner of the slide.

Réalisé par :
ZIDAN Loubna

Table des matières

1.	Introduction	2
a.	Qu'est-ce que la segmentation d'images ?	2
b.	U-NET pour la segmentation d'images :	2
c.	Défis des applications d'imagerie biomédicale :	2
2.	L'architecture U-Net :	3
3.	Augmentation de données :	4
4.	Expériences	5
a.	Définir le réseau :	6
b.	Trainer le modèle :	6
c.	Tester les performances :	7
d.	Définition du seuil de recherche de cellules :	7
5.	Problèmes que U-Net résout :	9
6.	Conclusion :	9

1. Introduction

Les réseaux de neurones convolutionnels (CNN, de l'anglais Convolutional Neural Network) sont à ce jour les modèles les plus performants pour la classification, la localisation et la segmentation des images surtout dans le domaine médical.

U-Net est un modèle issu du réseau de neurone convolutif traditionnel développé pour la segmentation des images biomédicales. Il permet la localisation et la segmentation en effectuant la classification sur chaque pixel.

a. Qu'est-ce que la segmentation d'images ?

La segmentation d'images extrait des informations précieuses des données visuelles tout en utilisant des algorithmes et des architectures d'apprentissage automatique hautement spécialisés pour donner un sens aux données afin d'identifier, distinguer, trier, faire des prédictions ou répliquer les données visuelles.

La segmentation d'images implique la fragmentation de toute forme visuelle de données en fragments délimités spécifiques représentant les classes ou objets appropriés pour en déduire des informations significatives à partir des données visuelles. Un processus pour identifier et classer des images, par exemple, des tables d'identification, une voiture, une personne, etc. applications, par exemple, détection de tumeurs, cancer, détection d'objets, systèmes de contrôle de la circulation, surveillance vidéo et imagerie biomédicale compréhension de données visuelles. La majorité de ces projets de vision par ordinateur impliquent la segmentation d'images comme première étape.

Pour répondre à de nombreuses questions biologiques, il est nécessaire de segmenter les cellules individuelles en images. La segmentation permet :

- ✓ calcul du nombre de cellules dans une image, utile par exemple pour comparer les effets des médicaments sur la survie des cellules
- ✓ Estimation précise des formes de cellules, également utile pour les mêmes raisons
- ✓ Mesures temporelles de la dynamique cellulaire telles que la division cellulaire, les mouvements cellulaires ou l'influx de calcium
- ✓ Quantification de l'expression de protéines ou d'ARN

b. U-NET pour la segmentation d'images :

U-Net est l'une des architectures de segmentation d'images les plus connues proposées en 2015 par Olaf Ronneberger, Philipp Fischer, Thomas Brox (Université de Fribourg, Allemagne). Il s'agit d'une technique de segmentation de bout en bout, ce qui signifie qu'elle prend une image brute et génère une carte de segmentation définie de l'image. Il s'agit d'une architecture Deep Convolutional Network spécialement conçue pour la segmentation des applications d'imagerie biomédicale. U-Net est capable de localiser l'image en prédisant l'image pixel par pixel.

c. Défis des applications d'imagerie biomédicale :

Dans le domaine de l'annotation d'images biomédicales, nous avons toujours besoin d'experts, qui ont acquis les connaissances associées, pour annoter chaque image. Et ils prennent également beaucoup de temps pour annoter. Si le processus d'annotation devient automatique, moins d'efforts humains et un coût moindre peuvent être obtenus. Ou cela

Chaque case bleue correspond à une carte d'entités multicanaux. Le nombre de canaux est indiqué en haut de la case. La taille x-y est fournie sur le bord inférieur gauche de la boîte. Les cases blanches représentent les cartes d'entités copiées. Les flèches indiquent les différentes opérations.

Avant d'approfondir l'architecture, définissons certains termes et concepts importants des réseaux convolutifs qui seront abordés sous peu :

- ❖ Taille du noyau (kernel size) : C'est le champ de vision de la convolution de l'image. Il s'agit essentiellement de la taille du masque filtrant.
- ❖ Max-pooling : il réduit les dimensions des caractéristiques afin d'extraire les caractéristiques fortement pondérées qui deviennent faciles à traiter, à calculer et nécessitent moins de puissance de calcul pour l'apprentissage des paramètres.
- ❖ Epoch : correspond à un apprentissage sur toutes les données, plus ce nombre est grand plus on devrait obtenir une bonne précision, mais bien sûr plus c'est long.
- ❖ taux d'apprentissage (Learning rate) : est un hyperparamètre qui joue sur la rapidité de la descente de gradient : un nombre d'itérations plus ou moins important est nécessaire avant que l'algorithme ne converge, c'est-à-dire qu'un apprentissage optimal du réseau soit réalisé.

Ce modèle fameux de la forme U (figure 1) est symétrique et se compose de deux parties principales : La partie gauche est appelée chemin de contraction de sous-échantillonnage (encodeur), il est utilisé pour capturer le contexte d'une image constituée de nombreux blocs de contraction. Chaque bloc applique deux couches de convolution 3x3 suivies d'une fonction Relu et d'un max pooling 2x2. Le nombre de filtres ou de mappages de caractéristiques après chaque bloc double afin que l'architecture puisse apprendre efficacement les structures complexes. La partie droite est un chemin expansif de suréchantillonnage (décodeur), il permet une localisation précise grâce à la convolution transposée, constitué de couches convolutives 2d transposées (Le nombre de blocs d'expansion est identique au nombre de blocs de contraction).

En plus de ces blocs de souséchantillonnage et de suréchantillonnage, il possède des connexions de saut des blocs de souséchantillonnage vers les blocs de suréchantillonnage, ce qui lui permet de propager des informations locales plus précises vers les couches ultérieures.

L'image de sortie segmentée résultante est beaucoup plus petite que l'image d'entrée brute. Il s'agit en fait d'un assemblage de couches de convolution et de couches de «max pooling» permettant de créer une carte de caractéristiques d'une image et de réduire sa taille pour diminuer le nombre de paramètres du réseau.

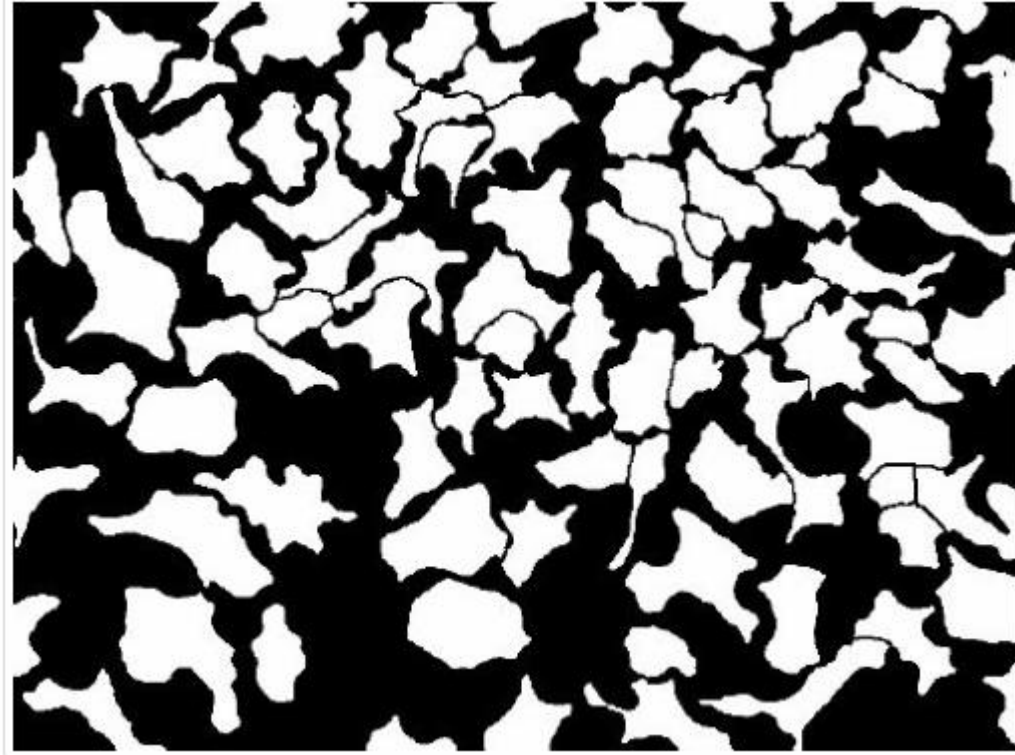
3. Augmentation de données :

L'augmentation de données consiste à augmenter la quantité de données disponibles pour un apprentissage. Cette augmentation capitalise sur cette base de qualité pour générer de nouvelles données tout aussi représentatives mais en amenant une variabilité supplémentaire. Cette variabilité renforce le caractère représentatif et exhaustif du jeu de données d'entraînement. À terme, cette augmentation de données permet de déboucher sur des modèles plus performants et fiables car plus robustes aux variations. Pour réaliser une transformation de données, il s'agira d'appliquer un traitement sur chaque image, pour générer de nouvelles images, des variantes de l'image initiale. On peut, par exemple, parler de rotation, de contraste, de rognage, etc.

Le dataset qu'on dispose contient très peu d'images, alors nous allons appliquer l'approche d'augmentation des données dans notre projet afin d'entraîner notre modèle sur un nombre plus grand de données.

4. Expériences

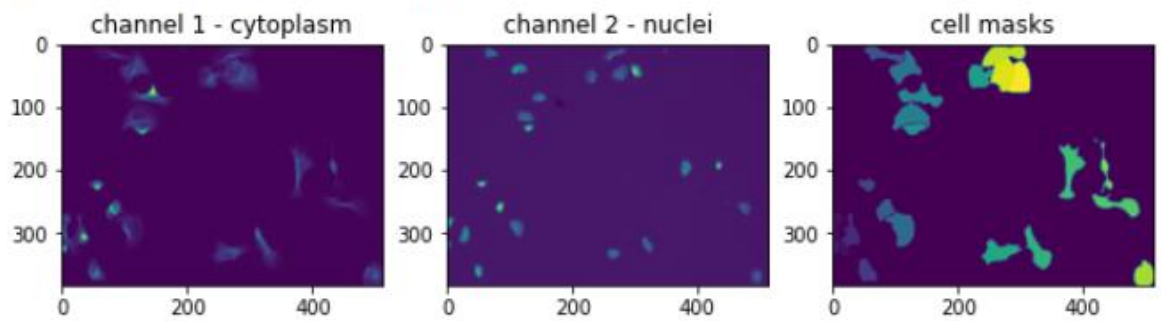
Les données d'apprentissage sont un ensemble de 30 images (512x512 pixels) de la microscopie électronique à transmission en coupe série du cordon nerveux ventral de la larve de premier stade de drosophile (VNC). Chaque image est accompagnée d'une carte de segmentation de vérité au sol entièrement annotée correspondante pour les cellules (blanc) et les membranes (noir).



La première partie de bloc-notes mettra en place un réseau convolutif U-net de base pour effectuer une segmentation cellulaire à l'aide d'une version organisée de cet ensemble de données, que la première cellule du bloc-notes téléchargera pour vous. Ce sont des images avec une coloration cytoplasmique (cellule entière colorée) et une coloration nucléaire (canaux 1 et 2 des images). Les masques de segmentation fournis sont destinés au cytoplasme (segmentation cellulaire entière). Il existe un code pour former et tester le réseau avec une simple fonction de coût.

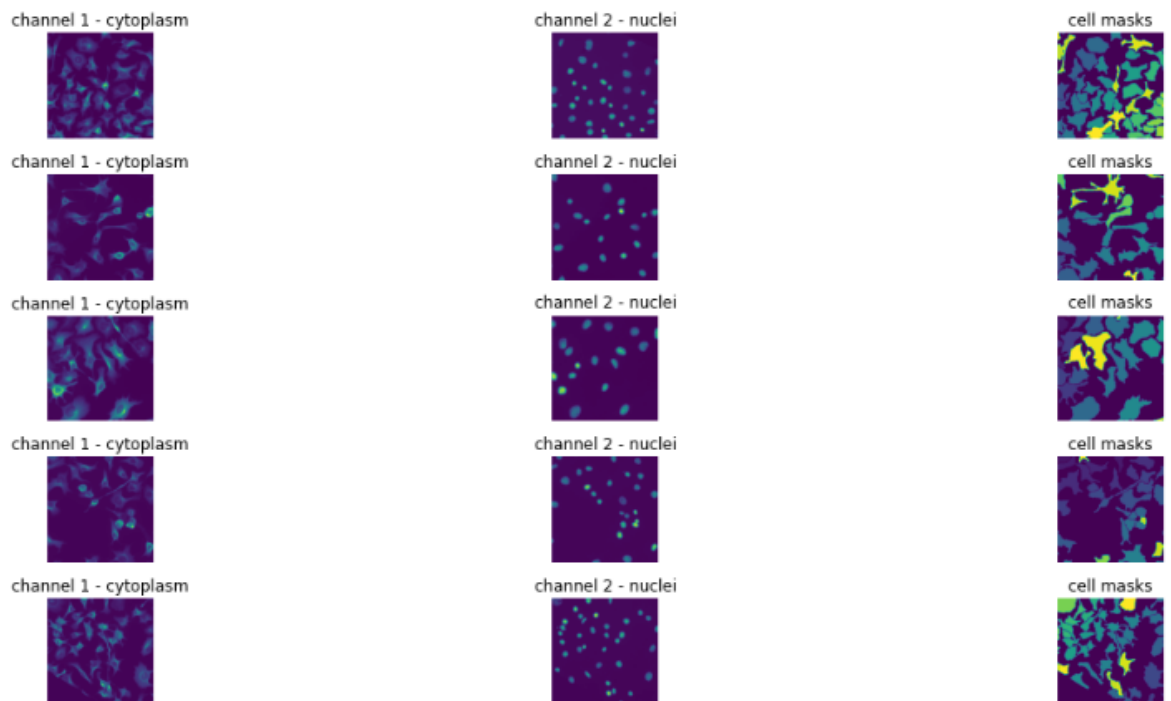
Après avoir télécharger et normaliser les données, on affiche une image d'apprentissage aléatoirement avec son mask :

there are 23 cells in this image



cell masks sont des étiquettes de 1, ... au nombre de cellules.

Ensuite, pour l'augmentation des données, on a une fonction de transformation pour les augmentations avec des paramétriques spécifiés aléatoirement. Et on a affiché 5 images avec leurs masks :



a. Définir le réseau :

On définit le réseau : `net = Unet(nbase, nout, kernel_size)` Avec `kernel-size=3`, `nbase = [2, 32, 64, 128, 256]` qui est le nombre de canaux par couche, et `nout = 2` qui est le nombre de sorties.

b. Trainer le modèle :

Les images d'entrée et leurs cartes de segmentation correspondantes sont utilisées pour entraîner le réseau avec l'implémentation de descente de gradient stochastique (SGD). En raison des convolutions non rembourrées, l'image de sortie est plus petite que l'entrée d'une largeur de bordure constante. Pour minimiser les frais généraux et utiliser au maximum la mémoire GPU, nous privilégions les grandes tuiles d'entrée sur une grande taille de lot et réduisons donc le lot à une seule image. En conséquence, nous

utilisons une impulsion élevée (0,99) telle qu'un grand nombre d'échantillons d'apprentissage vus précédemment déterminent la mise à jour dans l'étape d'optimisation actuelle. La fonction d'énergie est calculée par un soft-max pixel par pixel sur la carte de caractéristiques finale combinée avec la fonction de perte d'entropie croisée.

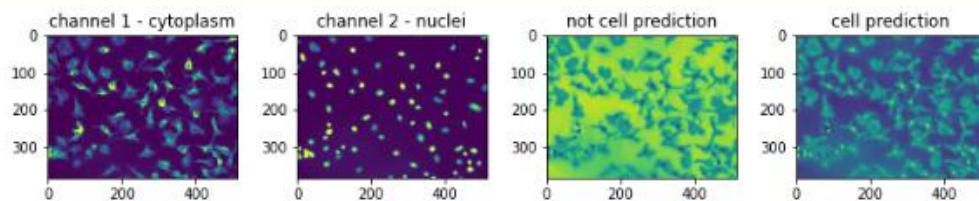
Cet apprentissage nécessite le réglage de certains paramètres comme le nombre d'époque, en fait, pour que l'apprentissage augmente linéairement en taille de lot, il faut augmenter `n_epochs` ! on la fixe à 30 époques.

Le taux d'apprentissage est resté comme sa valeur par défaut 0.1.

Ensuite on divise notre dataset en train et validation datasets et la fraction de données utiliser pour la validation est de 0.05.

c. Tester les performances :

Voyons comment le réseau fonctionne sur une image de test :

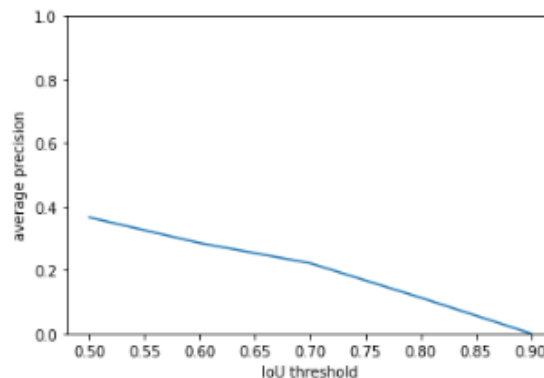


On voit bien que la dernière image à droite prédite très efficacement les cellules existantes ! mais on obtient deux images différentes pour les cellules prédites et pour les cellules non prédites, et nous aimerons bien attribuer chaque pixel d'une cellule à une cellule spécifique plutôt qu'à une classe générale

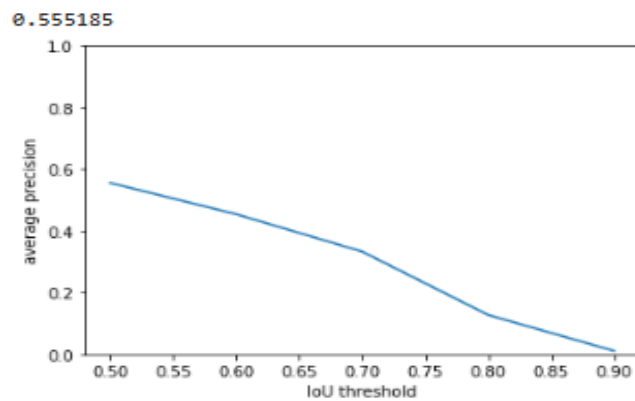
d. Définition du seuil de recherche de cellules :

Nous avons trouvé des zones de "pas de cellule" et de "cellule". Pour créer une segmentation d'instance, nous devons trouver un seuil qui produit les meilleures segmentations sur notre ensemble de validation. Comment définit-on une bonne segmentation ? Nous pouvons utiliser une mesure appelée intersection sur union (IoU) et appeler une cellule une bonne cellule si elle chevauche une cellule de vérité au sol avec un IoU supérieur à une certaine valeur. Nous avons pris le code de [cellpose/metrics.py] pour ce faire. Si on teste le modèle sur l'ensemble de validation, on obtienne une moyenne de précision de 36% avec une mesure de IOU de 90% :

0.36555332



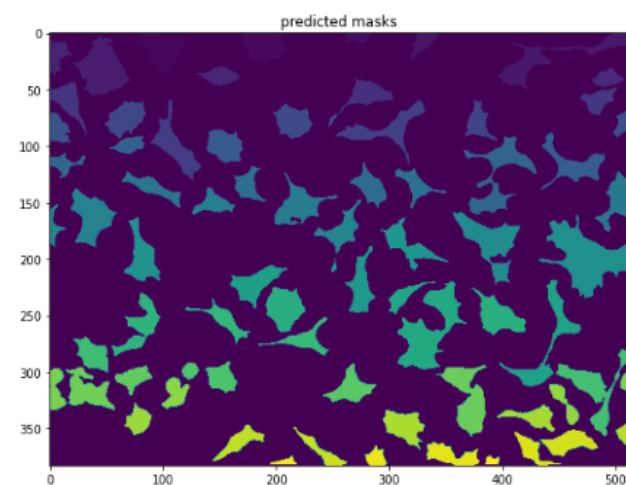
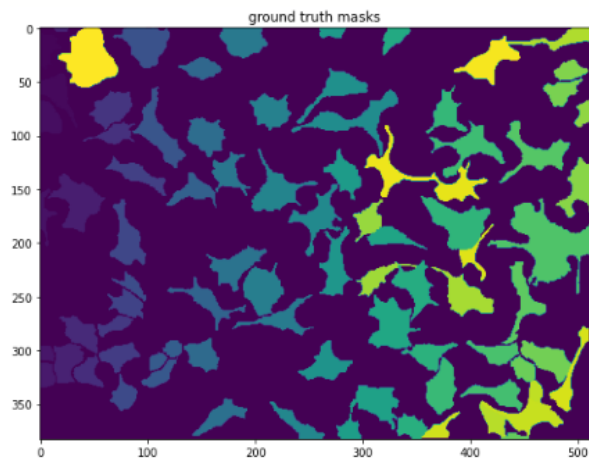
Une fois nous avons choisi un seuil, nous l'utilisons sur nos images de test, on reteste le modèle sur les images de test, et on obtient une moyenne de précision de 55% :



Seuil trouvé pour les données de validation 0.36

Seuil trouvé pour les données de test 0.55

On affiche la même image pour voir le resultat final après avoir déterminer un seuil, on obtient :



5. Problèmes que U-Net résout :

- ❖ Moins de données de formation : La faible disponibilité des données de formation en imagerie biomédicale peut nuire à la robustesse du système de segmentation. Pour résoudre ce problème et augmenter les données d'apprentissage, une propriété de déformation élastique aléatoire est utilisée. L'image résultante ressemble presque exactement à l'image d'origine et a également été correctement classée avec des déformations. C'est un avantage important de l'architecture U-Net.
- ❖ Objets tactiles : Objets tactiles de même classe qui doivent être correctement séparés. Les pixels de l'image d'arrière-plan sont insérés entre tous les objets en contact et un poids de perte individuel est attribué à chaque pixel. L'utilisation de la perte pondérée permet une forte pénalisation pour séparer les étiquettes de fond entre les cellules en contact.

Cette architecture U-Net a bien fonctionné pour des cas d'utilisation tels que la segmentation de structures neuronales en microscopie électronique, qui a été truffée de défis tels que les membranes floues, les structures à faible contraste.

Ainsi, dans l'ensemble, cette architecture est avantageuse pour la segmentation d'images et peut relever des défis tels que la faible disponibilité des données d'entraînement, les objets se touchant et se chevauchant, les frontières partiellement invisibles entre différents objets, les bordures floues, les bords à faible contraste et les objets avec de fortes variations de forme. Pour résoudre ces problèmes, U-Net est extrêmement avantageux pour les applications d'imagerie biomédicale, particulièrement en proie aux problèmes susmentionnés. Il a extrêmement bien fonctionné avec moins de données autour de 30 images annotées en tirant parti de sa précision pour exploiter la robustesse obtenue grâce à l'augmentation des données avec des déformations élastiques.

6. Conclusion :

L'architecture u-net atteint de très bonnes performances sur des applications de segmentation biomédicale. Grâce à l'augmentation des données avec des déformations élastiques, il ne nécessite que très peu d'images annotées et dispose d'un temps d'apprentissage très raisonnable.