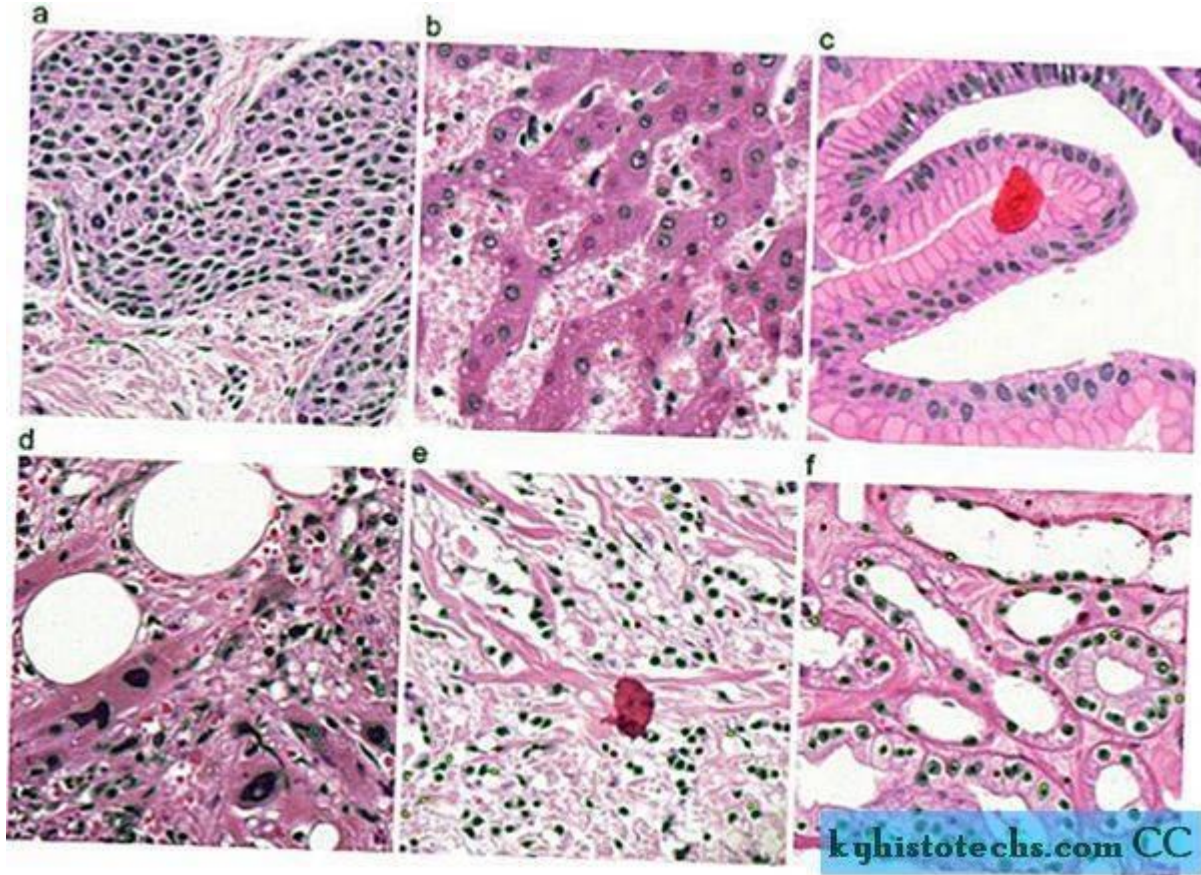


Segmentation-des-Images-de-la-Microscopie-lectronique



L'IA permet de simplifier et d'accélérer le travail des équipes d'imagerie, aujourd'hui majoritairement sur les étapes précliniques, en facilitant la lecture des images. Le CHRU a donc besoin d'un outil IA pour segmenter un tel type d'images. Il vous demande de l'aide pour cet objectif.

Contexte du projet

L'IA est aujourd'hui omniprésente dans la littérature scientifique de l'imagerie médicale, d'autant plus depuis le développement de nouveaux algorithmes appelés réseaux de neurones convolutifs.

En effet, à ce jour, l'IA est très utile dans le domaine de l'imagerie, sur deux volets : la classification des images et la segmentation des organes. Les algorithmes pour classer les images peuvent permettre d'aider au diagnostic en classant une image dans une catégorie particulière de pathologie. Les algorithmes pour segmenter les images sont couramment utilisés sur tous les types d'imagerie et en routine au CHRU. C'est ainsi que l'IA permet un gain de temps aux praticiens à la fois pour le diagnostic ou lors d'interventions. Elle présente aussi l'avantage de contourner certains biais liés à l'interprétation de l'opérateur.

Les données.

Les données proviennent du défi ISBI 2012 pour la segmentation des structures neuronales dans l'empilement de microscopie électronique.

Les données de formation sont un ensemble de 30 sections d'une section série de données de microscopie électronique à transmission (ssTEM) du cordon nerveux ventral de la larve du premier stade de la drosophile. Le microcube mesure 2 x 2 x 1,5 microns environ, avec une résolution de 4 x 4 x 50 nm / pixel. Chaque image a 512x512 pixels.

Les étiquettes binaires correspondantes (répertoire label) sont fournies en blanc pour les pixels des objets segmentés et en noir pour le reste des pixels (qui correspondent principalement à des membranes).

L'objectif étant de confronter nos images train avec les masks (labels) afin de pouvoir effectuer notre modèle de prédictions.

Les données sont disponibles au format TIF ou TIFF (Tagged Image File Format) qui est un format de fichier graphique bitmap (raster).

Caractéristiques du format TIF

Le format TIFF est un ancien format graphique, permettant de stocker des images bitmap (raster) de taille importante (plus de 4 Go compressées), sans perte de qualité et indépendamment des plates formes ou des périphériques utilisés (Device-Independent Bitmap, noté DIB).

Le format TIFF permet de stocker des images en noir et blanc, en couleurs réelles (True color, jusqu'à 32 bits par pixels) ainsi que des images indexées, faisant usage d'une palette de couleurs. Ce format peut être utilisé pour des images médicales.

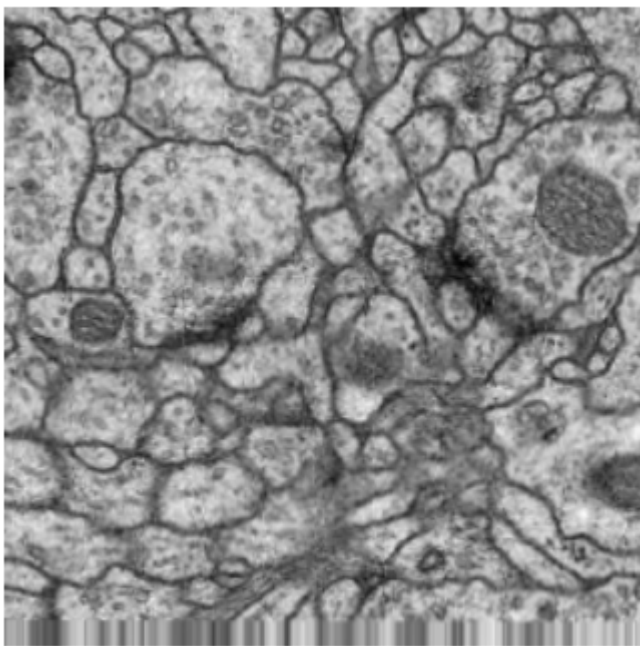
Nous utiliserons google colab pour le lancement de notre modèle.

Nous allons dans un premier temps transformer nos images .tiff en format .png pour pouvoir construire le modèle et pour pouvoir l'exploiter par la suite.

Visualisation de nos données train (jeu d'apprentissage)

Nous souhaitons établir ici une comparaison entre nos images et nos masks afin de pouvoir par la suite établir un modèle de prédiction sur nos images de test. Nous allons donc dans un premier temps visualiser nos images ainsi les masks associés sur le jeu d'entraînement.

Image 0 du jeu d'entraînement



Masque associé à l'image 0



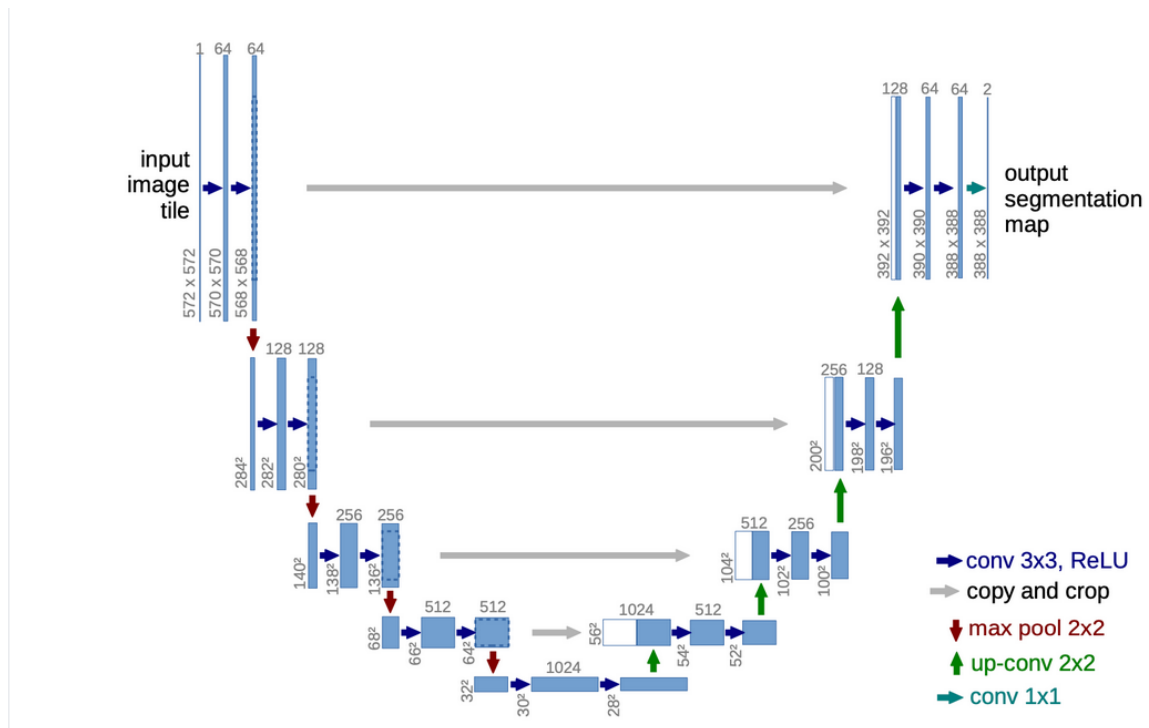
Nous avons ici l'image 0 avec son mask associé.

Architecture du modèle: Segmentation grâce au réseau neuronal U Net

Pour réaliser la segmentation de nos images microscopiques, nous utiliserons U-Net qui est un réseau neuronal entièrement convolutif avec une structure codeur-décodeur conçu pour la segmentation d'images sementiques sur des images biomédicales. C'est une architecture de méta-réseau très efficace qui a été adaptée pour incorporer d'autres conceptions d'architecture de réseau neuronal convolutif.

L'architecture U-net est synonyme d'architecture encodeur-décodeur. Il s'agit essentiellement d'un cadre d'apprentissage en profondeur basé sur les FCN; il comprend deux parties:

Un chemin contractuel similaire à un encodeur, pour capturer le contexte via une carte de caractéristiques compacte. Un chemin d'extension symétrique similaire à un décodeur, qui permet une localisation précise. Cette étape est effectuée pour conserver les informations de limite (informations spatiales) malgré le sous-échantillonnage et le regroupement maximal effectués dans l'étape du codeur.



Architecture U-net (exemple pour 32x32 pixels dans la résolution la plus basse).

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

Avantages de l'utilisation d'U-Net

1. Efficace en termes de calcul
2. Entraînable avec un petit ensemble de données
3. Formé de bout en bout
4. Préférable pour les applications biomédicales

Chaque pixel d'une image est doté d'une étiquette de classe. Vous obtenez des cartes de segmentation, qui ressemblent à celle de la figure ci dessus. Auparavant, les chercheurs biomédicaux suivaient deux approches:

1. Classifier l'image dans son ensemble (maligne ou bénigne).
2. Diviser les images en patches et les classer.

En raison de l'augmentation de la taille de l'ensemble de données, la correction était certainement meilleure qu'une classification d'image entière, cependant, il y avait quelques inconvénients liés à la même chose. De plus petits pas ou des correctifs avec beaucoup de chevauchement sont à la fois intensifs en calcul et entraînent des informations redondantes (répétitives). De plus, un bon compromis entre les informations contextuelles et la

localisation est vital. Les petits correctifs entraînent une perte d'informations contextuelles tandis que les gros correctifs altèrent les résultats de la localisation. Enfin, les correctifs qui ne se chevauchent pas entraînent une perte d'informations de contexte. Sur la base d'observations antérieures, une architecture codeur-décodeur produit des valeurs d'intersection sur union (IoU) beaucoup plus élevées que l'alimentation de chaque pixel à un CNN pour la classification.

Augmentation des données

Les données pour la formation ne sont pas suffisantes pour alimenter un réseau de neurones d'apprentissage en profondeur. On va utiliser un module appelé `ImageDataGenerator` dans `keras.preprocessing.image` pour augmenter les données.

Les données sont rares dans le domaine de l'imagerie médicale (seulement 30 images supervisées dans cet ensemble de données); cependant, les réseaux neuronaux reposent souvent sur une grande quantité de données supervisées pour obtenir de bons résultats; par conséquent, l'augmentation des données est largement utilisée. On peut proposer, non seulement la transformation affine typique telle que la translation, la rotation et le recadrage, mais aussi l'utilisation de la déformation élastique. La déformation est une technique largement adoptée pour les tâches de segmentation d'images biomédicales, car des objets comme les cellules et les organes ont souvent des structures non rigides.

Les transformations affines et la déformation élastique sont appliquées simultanément aux images et aux étiquettes. Les réglages de luminosité et le bruit gaussien sont ajoutés aux images d'entrée.

Pendant l'entraînement, toutes les augmentations sont choisies de manière stochastique; pour chaque image et étiquette, l'augmentation est un composite de différentes combinaisons de transformations. Pour la déformation élastique, le paramètre α est choisi entre 100 et 300, et le paramètre σ est choisi entre 10 et 15.

`keras.preprocessing.image.ImageDataGenerator` est un générateur de données, qui peut alimenter le DNN avec des données telles que: (données, étiquette), il peut également augmenter les données en même temps.

Il est très pratique pour nous d'utiliser `keras.preprocessing.image.ImageDataGenerator` pour augmenter les données en implémentant la rotation d'image, le décalage, la redimensionnement, etc.

Pour les tâches de segmentation d'image, l'image et le masque doivent être transformés ensemble.

Une fois créé le générateur de données et la visualisation du résultat issu de la `Data Augmentation`, nous allons pouvoir définir notre modèle.

Nous obtenons une classe de 30 images.

On définit dans un premier temps la fonction de perte (loss).

Fonction de perte: Loss Function

Puisqu'il s'agit d'une tâche de segmentation segmentaire, une perte de pixels est calculée via une fonction softmax combinée à une entropie croisée sur la carte de caractéristiques finale.

Les images médicales contiennent souvent des classes d'objets très déséquilibrées; par conséquent, il est préférable de suggérer l'utilisation d'une perte pondérée. La fonction de pondération intègre également la distance aux deux cellules les plus proches et est définie par la formule suivante.

$$w(x) = w_c(x) + w_0 \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right)$$

Définition du modèle

Le modèle a été formé sur 90% des données d'entraînement (27 images) et testé sur 10% des données (3 images) avec les hyperparamètres suivants:

Époque: 50

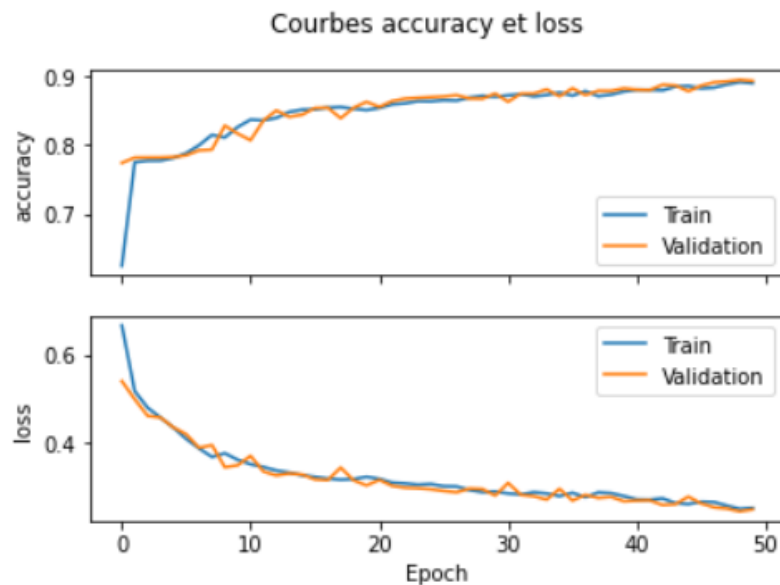
Optimiseur: Adam

Fonction objective: perte d'entropie croisée pondérée par pixel "binary_crossentropy":
Calcule la perte d'entropie croisée entre les étiquettes vraies et les étiquettes prévues.

On va créer des fichiers npy contenant toutes les images et masques. Nous allons entraîner notre modèle sur le fichier npy.

imgs_train correspond à notre feature(X) (images) tandis que **imgs_mask_train (mask)** correspond à notre target(Y).

Affichage des courbes de loss et d'Accuracy sur les données d'apprentissage et de test

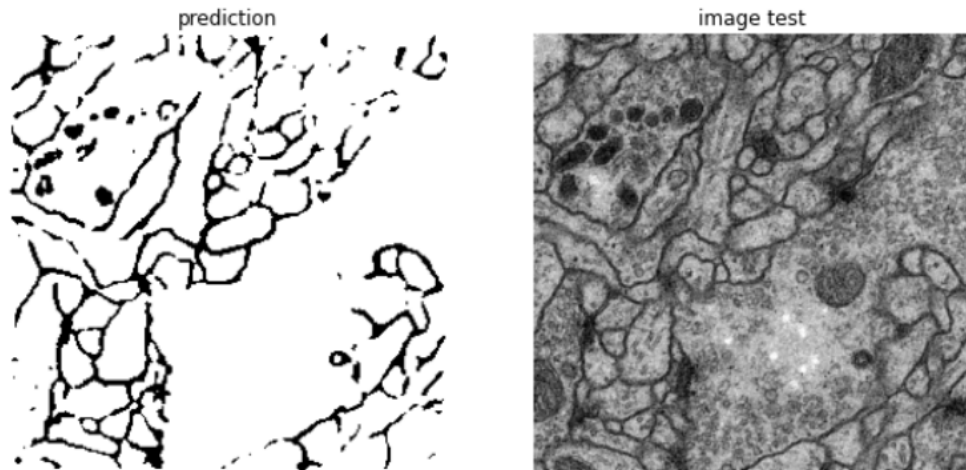


Nous obtenons près de 90% d'accuracy pour notre modèle que ce soit sur le train ou la validation. On observe une stagnation au niveau de l'accuracy et du loss pour notre modèle d'entraînement à partir de 30 époques, et le modèle atteint des performances de test maximales à la 50e époque. Différentes tailles de lots et taux d'apprentissage ont été expérimentés pour entraîner le modèle jusqu'à 100 époques. La perte d'entraînement diminue mais n'améliore pas les performances de segmentation. Le modèle est probablement en surentraînement.

On effectue par la suite les prédictions du modèle sur le jeu de test et on enregistre des résultats prévus.

visualisation des prédictions

Voici un exemple de comparaison entre une image du jeu de test et sa prédiction.



Conclusion

L'architecture u-net atteint de très bonnes performances sur des applications de segmentation biomédicale très différentes. 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. L'architecture u-net peut être appliquée facilement à de nombreuses autres tâches