Microsoft en France
Ecole IA
ISEN ALL IS DIGITAL! BREST
yncréa
SIMPLON.CO

Septembre 2021

Reconnaissance automatique des anomalies chromosomiques afin d'accélérer le diagnostic pronostic dans la prise en charge des cancers du sang (Part I)

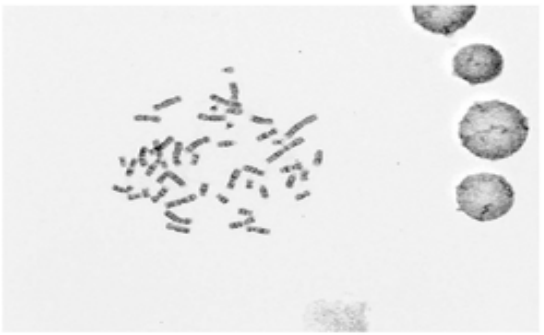
Amaury, Aude, Jamal, Luigi

SOMMAIRE:

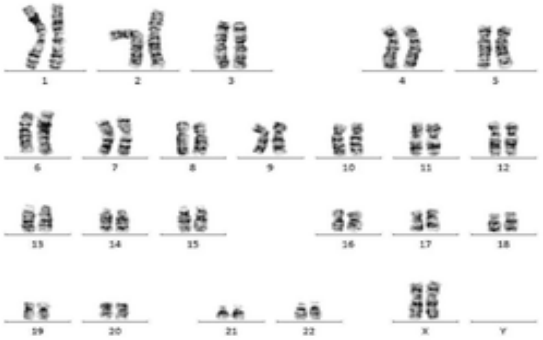
Introduction:

Les chromosomes sont des éléments microscopiques constitués par la molécule d'ADN semblable à une chaîne. Ils sont à l'intérieur du noyau de chaque cellule. Ils portent des informations génétiques telles que la couleur des cheveux, la couleur des yeux et la taille. Tout être humain en bonne santé possède 23 paires de chromosomes au total, 22 paires sont constitués de chromosomes non sexuels (autosomes) et la 23ème paire correspond aux chromosomes sexuels (gonosomes).

L'analyse du nombre et de l'apparence des chromosomes fournit des informations pour dépister les troubles génétiques d'une personne causés par des anomalies génétiques. Pour pouvoir mettre en évidence ce type d'anomalies chromosomiques (par exemple une trisomie), il est donc nécessaire de classer les chromosomes afin d'établir le caryotype, puis de l'analyser.



Une image microscopique colorée au Giemsa en métaphase



Résultat du caryotypage de chromosomes appariés et ordonnés.

Problématique:

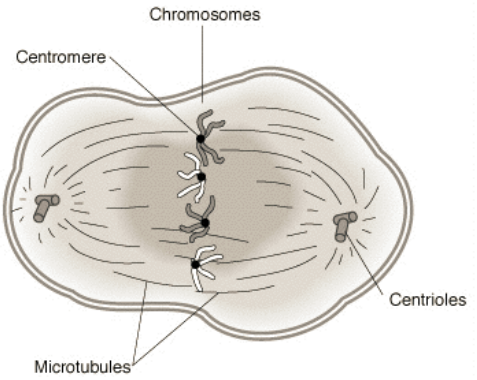
En cytogénétique, les expériences commencent généralement à partir de préparations chromosomiques fixées sur des lames de verre. Parfois, un chromosome peut tomber sur un autre, produisant des chromosomes qui se chevauchent dans l'image. Avant les ordinateurs et le traitement des images avec la photographie, les chromosomes étaient découpés à partir d'une image papier puis classés (au moins deux images papier étaient nécessaires lorsque les chromosomes se chevauchaient). Plus récemment, des méthodes de segmentation automatique ont été développées pour surmonter ce problème. La plupart du temps, ces méthodes reposent sur une analyse géométrique du contour chromosomique et nécessitent une intervention humaine en cas de chevauchement partiel. Les techniques modernes de Deep Learning ont le potentiel de fournir une solution plus fiable et entièrement automatisée.

Une solution de segmentation rapide et entièrement automatisée peut permettre d'étendre certaines expériences à un très grand nombre de chromosomes, ce qui n'était pas possible auparavant.

Partie I : Segmentation des images des chromosomes

Ce projet est composé de deux parties, la première que nous traitons ici se charge de la segmentation et de la classification. La deuxième partie sera consacrée à la détection d'anomalies.

La métaphase correspond à un stade de la mitose (division cellulaire) durant laquelle les chromosomes sont condensés à l'équateur de la cellule avant d'être séparés dans ce qui deviendra deux cellules distinctes. Durant cette étape, les chromosomes sont donc bien visibles car c'est là qu'ils sont les plus condensés. On utilise donc cette phase pour réaliser les caryotypes.



chromosomes en métaphase

Malheureusement, les images microscopiques de chromosomes en métaphase ne sont pas directement exploitables pour réaliser un caryotype car les chromosomes y sont en vrac et peuvent présenter des chevauchements. C'est le cas avec notre dataset.

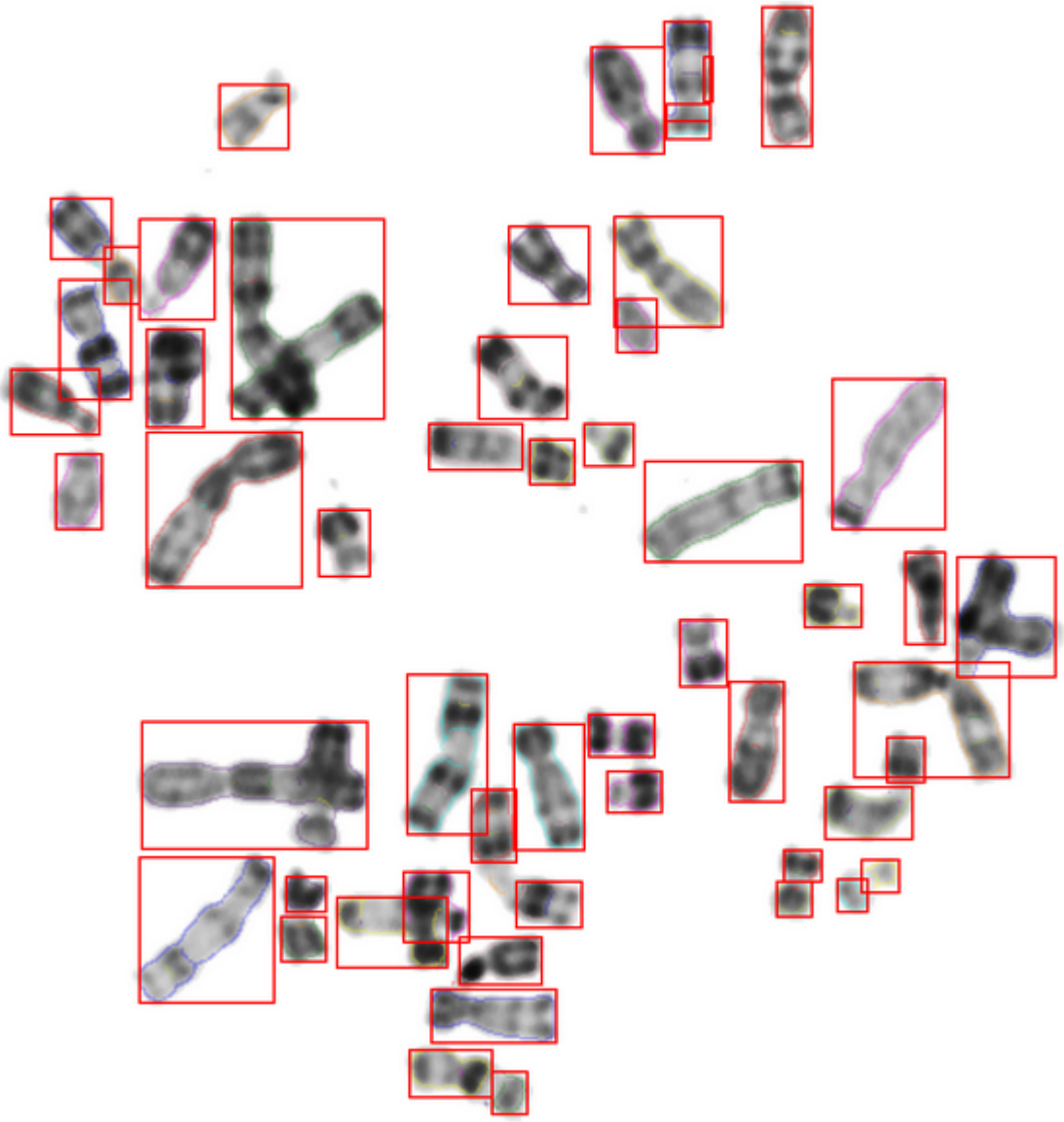
Il est donc nécessaire de séparer ces chromosomes pour pouvoir les replacer séparément sur un caryotype.

Propriétés de régions avec skimage:

Tout d'abord, il s'agissait de récupérer les amas de chromosomes sur l'image globale. Pour cela nous avons utilisé la librairie skimage (scikit image) et son outil regionprops, qui permet de récupérer les propriétés de régions de l'image.

L'idée est d'utiliser le contraste qui existe entre le fond de l'image et les amas de chromosomes eux-mêmes. Un seuil de taille de région est fixé (pour éviter de récupérer des petites "taches" sur l'image).

Nous récupérons les coordonnées de chaque région et récupérons ainsi les paquets de chromosomes.



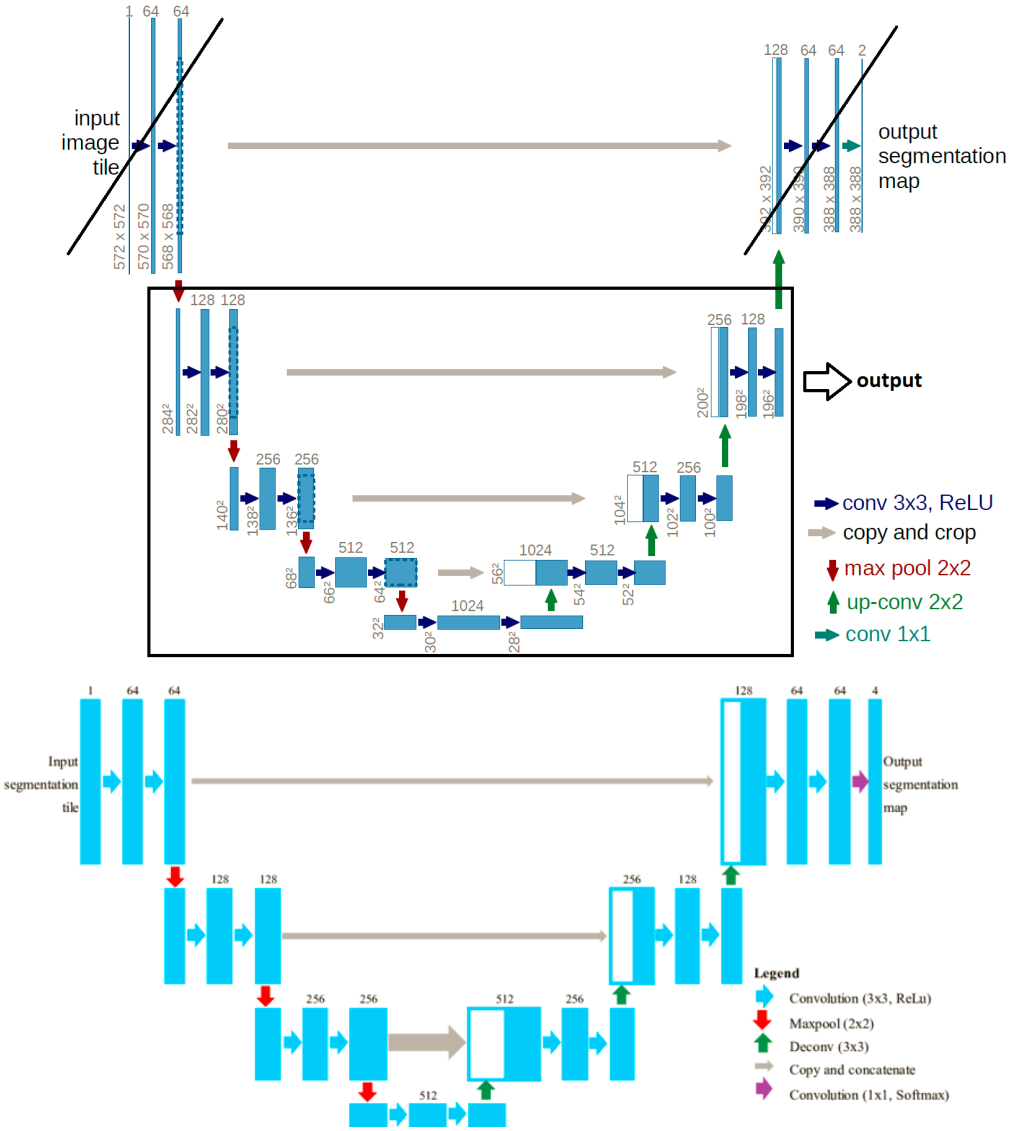
Le réseau U-net:

Le U-net est une technique de segmentation d'images développée principalement pour l'analyse d'images médicales qui peut segmenter avec précision des images en utilisant une quantité limitée de données d'entraînement. Ces caractéristiques confèrent à U-net une très grande utilité au sein de la communauté de l'imagerie médicale et ont entraîné une large adoption de U-net comme outil principal pour les tâches de segmentation en imagerie médicale. Le succès de U-net est évident dans son utilisation généralisée dans toutes les principales modalités d'imagerie, des tomodensitogrammes et IRM aux rayons X et à la microscopie.

Architecture U-net de base:

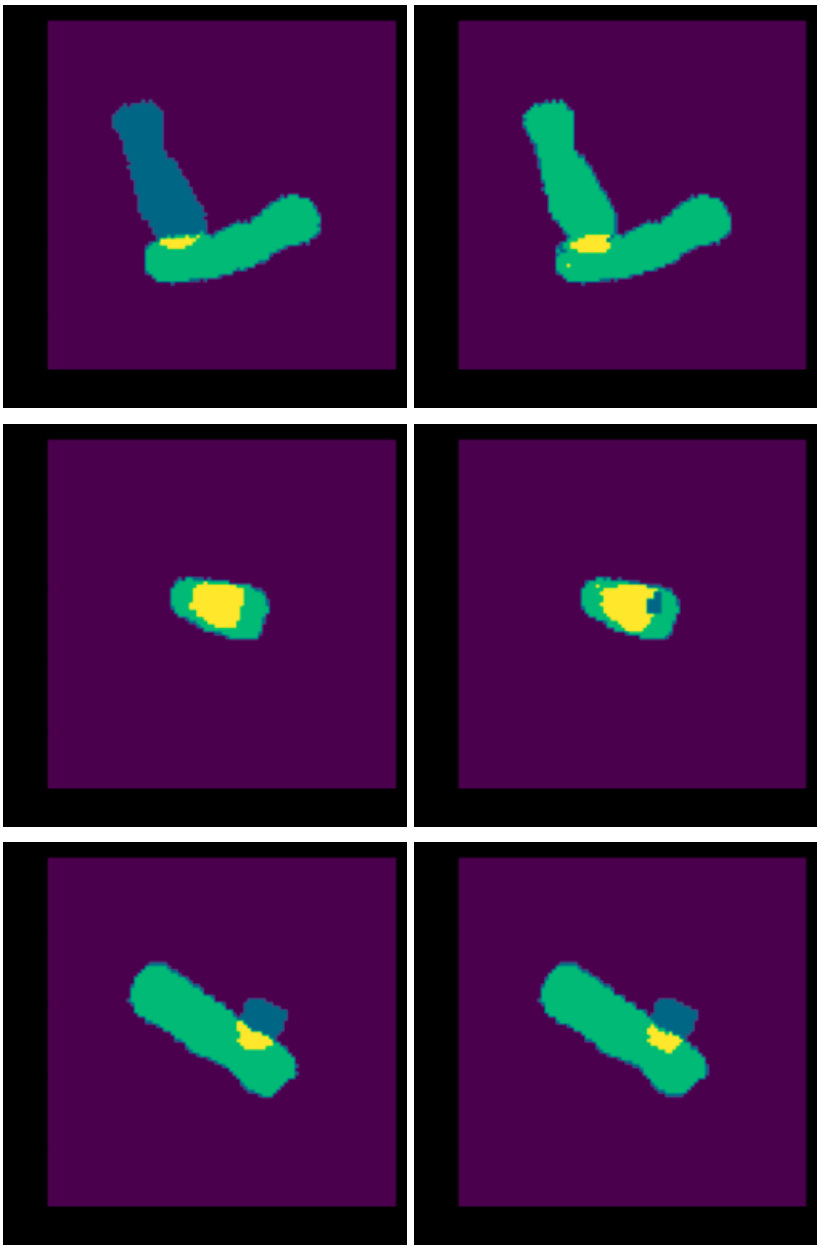
Les flèches représentent les différentes opérations, les cases bleues représentent la carte des caractéristiques de chaque couche et les cases grises représentent les cartes des caractéristiques recadrées du chemin de contraction.

Avant d'opter pour l'approche avec Pytorch, nous avons réalisé un premier essai de segmentation sur Tensorflow à l'aide d'un U-Net raccourci par rapport au U-net classique.



Architecture U-net classique tronquée issue du papier "Overlapping Chromosome Segmentation using U-Net: Convolutional Networks with Test Time Augmentation"

L'approche était prometteuse et offrait une bonne précision sur le papier (score de coefficient dice de 98%). Dans les faits et sur les images segmentées obtenues, malgré les 98% de précision, certaines prédictions étaient quasi parfaites mais d'autres avaient des scories de classes mal prédites aux extrémités des chromosomes voire parfois des chromosomes entiers mal prédits.



Exemples (test à gauche, prédiction à droite)

Quelques changements sur l'architecture ne permettaient pas d'améliorer sensiblement ce résultat. Nous avons émis l'hypothèse que c'était peut-être dû à un déséquilibre dans la représentation des classes car il y a beaucoup plus de pixels ayant la classe 0 correspondant au fond que de classes utiles pour les chromosomes et leur chevauchement ce qui peut nuire à l'apprentissage. Cependant, nous n'avons pas pu optimiser davantage le modèle pour avoir un

meilleur résultat faute de temps et à cause de difficultés techniques pour appliquer un poids à ces classes dans le cas d'un U-net de prédiction multiclasse.

```
Model: "model_4"
```

Layer (type)	Output Shape	Param #	Connected to
input_4 (InputLayer)	[(None, 96, 96, 1)]	0	
conv2d_24 (Conv2D)	(None, 96, 96, 64)	640	input_4[0][0]
batch_normalization_22 (BatchNo	(None, 96, 96, 64)	256	conv2d_24[0][0]
conv2d_25 (Conv2D)	(None, 96, 96, 64)	36928	batch_normalization_22[0][0]
batch_normalization_23 (BatchNo	(None, 96, 96, 64)	256	conv2d_25[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 48, 48, 64)	0	batch_normalization_23[0][0]
dense_1 (Dense)	(None, 48, 48, 64)	4160	max_pooling2d_5[0][0]
conv2d_26 (Conv2D)	(None, 48, 48, 128)	73856	dense_1[0][0]
batch_normalization_24 (BatchNo	(None, 48, 48, 128)	512	conv2d_26[0][0]
conv2d_27 (Conv2D)	(None, 48, 48, 128)	147584	batch_normalization_24[0][0]
batch_normalization_25 (BatchNo	(None, 48, 48, 128)	512	conv2d_27[0][0]
max_pooling2d_6 (MaxPooling2D)	(None, 24, 24, 128)	0	batch_normalization_25[0][0]
conv2d_28 (Conv2D)	(None, 24, 24, 256)	295168	max_pooling2d_6[0][0]
batch_normalization_26 (BatchNo	(None, 24, 24, 256)	1024	conv2d_28[0][0]
conv2d_29 (Conv2D)	(None, 24, 24, 256)	590080	batch_normalization_26[0][0]
batch_normalization_27 (BatchNo	(None, 24, 24, 256)	1024	conv2d_29[0][0]
conv2d_transpose_4 (Conv2DTrans	(None, 48, 48, 128)	131200	batch_normalization_27[0][0]
concatenate_4 (Concatenate)	(None, 48, 48, 256)	0	conv2d_transpose_4[0][0] batch_normalization_25[0][0]
conv2d_30 (Conv2D)	(None, 48, 48, 128)	295040	concatenate_4[0][0]
batch_normalization_28 (BatchNo	(None, 48, 48, 128)	512	conv2d_30[0][0]
conv2d_31 (Conv2D)	(None, 48, 48, 128)	147584	batch_normalization_28[0][0]
batch_normalization_29 (BatchNo	(None, 48, 48, 128)	512	conv2d_31[0][0]
conv2d_transpose_5 (Conv2DTrans	(None, 96, 96, 128)	65664	batch_normalization_29[0][0]
concatenate_5 (Concatenate)	(None, 96, 96, 192)	0	conv2d_transpose_5[0][0] batch_normalization_23[0][0]
conv2d_32 (Conv2D)	(None, 96, 96, 64)	110656	concatenate_5[0][0]
batch_normalization_30 (BatchNo	(None, 96, 96, 64)	256	conv2d_32[0][0]
conv2d_33 (Conv2D)	(None, 96, 96, 64)	36928	batch_normalization_30[0][0]
batch_normalization_31 (BatchNo	(None, 96, 96, 64)	256	conv2d_33[0][0]
conv2d_34 (Conv2D)	(None, 96, 96, 4)	260	batch_normalization_31[0][0]

```
Total params: 1,940,868
Trainable params: 1,938,308
Non-trainable params: 2,560
```

Sommaire de notre premier essai de modèle de segmentation

Finalement, nous avons opté pour une solution utilisant PyTorch, depuis un GitHub existant :
https://github.com/HKU-BAL/ChromSeg.git

La solution est en deux parties: premièrement, séparer les chromosomes, le fond et la partie de chevauchement.

Puis, à partir de cela, reconstituer les chromosomes individuels.

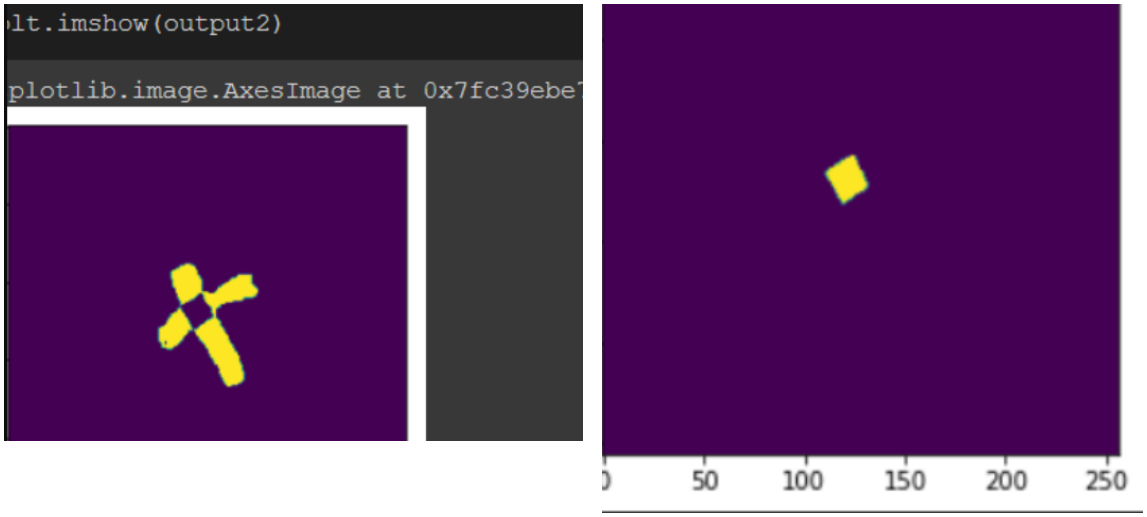
La première partie du U-Net est appelée l'encoder, elle est composée de 4 double couches de convolution (composées chacune d'une couche conv2D, d'une couche de normalisation et d'une activation reLU). L'encoder réalise l'extraction de feature sur les chromosomes et rétrécit l'image au fur et à mesure des convolutions.

La deuxième partie du U-Net réalise l'upsampling pour retourner à la taille initiale des images. Elle contient des couches de conv2D transpose qui réalisent la mise à l'échelle.

Entre les deux parties du U, il y a le mid ou skip Layer qui réinjecte l'information spatiale issue du downsampling dans le upsampling.

Durant le upsampling dans la phase d'expansion avec les conv transpose 2D, l'information spatiale recrée est imprécise. Le U-Net utilise donc les skip connections qui combinent l'information spatiale issue du downsampling avec celles du upsampling. Cependant, ce fonctionnement introduit des redondances dans l'extraction de features car la représentation des features est de faible qualité dans les premières couches du downsampling. Le mécanisme d'attention (douce) est implémenté dans les skip connexions ce qui va supprimer l'activation dans les régions non pertinentes. Ainsi, la redondance des features transférée entre les deux branches du U-Net est réduite et le modèle se concentre sur les activations importantes pour utiliser moins de puissance de calcul et surtout avoir une meilleure généralisabilité.

Résultats:



après reconstitution:

Partie II : Classification des images des chromosomes

L'analyse du caryotype est importante pour le diagnostic, le traitement et pronostic de maladies telles que les malformations congénitales et les tumeurs hématologiques. L'identification des chromosomes et leurs variations de structure par rapport aux images de métaphase en bande G est un élément important du processus de caryotypage, et est également le plus difficile.

La classification automatique des chromosomes devient urgente puisque de plus en plus d'échantillons de patients sont soumis à un examen médical tel qu'une biopsie de la moelle osseuse. Avec le développement de l'intelligence artificielle, les réseaux de neurones convolutifs (CNN) ont montré de bonnes performances en reconnaissance d'images.

Dans cette étude, à l'aide de l'ensemble de données étiquetées pour classer les chromosomes en 24 classes, nous allons utiliser une architecture en opensource (apache 2.0) de type Faster-RCNN, inventée par le laboratoire de recherche de Facebook et utilisant leur bibliothèque, detectron2.

Il y a deux grands ensembles dans l'architectures, un FPN (feature pyramid network) et un RPN (Region proposal network).

Le FPN est en deux parties, une première partie ressemble à un CNN classique avec des couches de convolution successives qui terminent par un maxpooling. Il y a ensuite la partie resnet qui contient les bottleneck blocks. Ces blocs contiennent des couches de convolution et de normalisation et ont un rôle de réduction de dimensionnalité pour diminuer le temps de calcul par le modèle.

Le RPN génère des propositions pour la région où se trouve l'objet. La proposition est un petit réseau qui se place par-dessus la feature map c'est-à-dire l'output de la couche convolutionnelle précédente. Le RPN a un classificateur et un régresseur. Il y a aussi le concept d'anchors (ancres). L'anchor est au centre de la proposition qui glisse sur la feature map. Le classificateur détermine la probabilité qu'une proposition contienne l'objet. Le régresseur ressort les coordonnées de la proposition.

La dernière couche du modèle est un conv2D appelé le prédicteur qui ressort la classe du chromosome.

Le réseau réalise de la data-augmentation durant son train à l'aide d'un module de detectron2, cela permet d'avoir une bonne accuracy en validation malgré la frugalité des données qui lui sont passées en train (nous n'avons qu'une cinquantaine d'échantillons par classe).

Résultats:

Le classificateur a donné une précision de 93,79 % pour l'identification des chromosomes. Le résultat a démontré que le RCNN a un potentiel dans la classification des chromosomes et contribuera à la construction d'une plateforme de caryotypage automatique.

Conclusion

Les résultats que nous avons obtenus sont encourageants, mais nous espérons les faire évoluer à travers les résultats des travaux futurs de la communauté open source et des chercheurs.

Nous pensons également que certains points pourrait évoluer comme :

- L'ensemble de données qui peut être complété par des images de chromosomes uniques et de plus de deux chromosomes qui se chevauchent.
- L'augmentation des données (data aug.) peut également inclure des transformations telles que des rotations et des étirements.
- Des hyperparamètres supplémentaires peuvent également être explorés, tels que les poids d'échantillon, le nombre de filtres et le nombre de couches. L'augmentation de la taille de la convolution pourrait éventuellement améliorer la classification erronée entre les chromosomes.
- Pour le suréchantillonnage, au lieu de recadrer les couches, le décodeur peut utiliser des indices de pooling calculés dans l'étape max-pooling de l'encodeur correspondant, comme dans Segnet.

Webographie:

Github :
A Two stage framework for crossing-overlap chromosome segmentation:
https://github.com/HKU-BAL/ChromSeg

Two-Stage Framework for Overlapping Chromosome Segmentation and Reconstruction : http://www.bio8.cs.hku.hk/pdf/chromseg.pdf

A Geometric Approach To Fully Automatic Chromosome Segmentation:
https://arxiv.org/pdf/1112.4164.pdf

Detectron Faster-RCNN par Facebook Research :
https://github.com/facebookresearch/detectron2

Overlapping Chromosome Segmentation using U-Net: Convolutional Networks with Test Time Augmentation :
https://www.researchgate.net/publication/336537768_Overlapping_Chromosome_Segmentation_using_U-Net_Convolutional_Networks_with_Test_Time_Augmentation