Architectures CNN pour la segmentation d'image

Félix Husson

5 décembre 2025

Table des matières

Chapitre 1

Architectures U-Net

1.1 Introduction et objectif

Ce document est un rapport sur les articles [2] et [3] dans le cadre du cours d'imagerie biomédicale du Master MMA de l'université Paris Cité. Mon but sera de présenter les points clés pertinents de ces articles et de mettre en œuvre un cas pratique 1.2 pour la segmentation d'image sur une base de données faite maison.

1.1.1 L'évolution d'une architecture de dee p learning

Dans cette section, nous allons voir comment l'évolution des architectures des réseaux à convolution pour la segmentation d'image. Historiquement, les algorithmes de segmentation d'image reposaient et étaient issus du domaine de la vision par ordinateur. C'est ainsi que les réseaux FCN [1] ont été construits, il s'agit de CNN classiques dont le but est de segmenter une image. Cependant, en imagerie biomédicale, on a moins de données d'une part et d'autre part, on souhaite une segmentation plus précise au niveau des pixels. C'est ainsi que l'architecture U-Net dérivé des CNN a été novatrice en faisant le compromis entre le contexte de l'image et la précision de localisation. Enfin, nous verrons les réseaux AG U-Net qui améliore les performances des réseaux U-Net classique en inhibant ou non des régions locales.

FCN

La figure 1.1 représente la structure du réseau FCN, qui prend en entrée une image, applique plusieurs convolutions et plusieurs échantillonnages successifs jusqu'à obtenir la classe de chacun des pixels de l'image. L'image en sortie est de la même taille que l'entrée et la valeur de chaque pixel est la classe à laquelle il appartient. Le canapé, le fond, la caméra, le chien, le chat ont toute une couleur différente puisqu'ils sont segmentés en différentes classe.

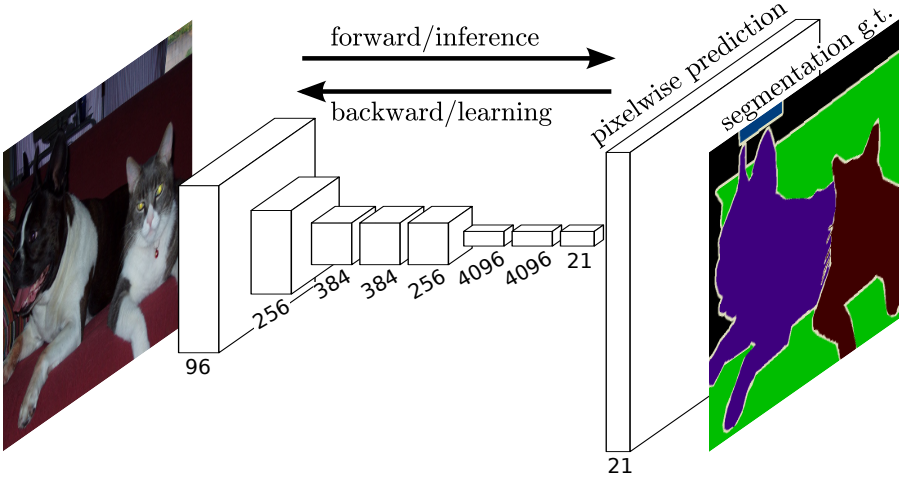

FIGURE 1.1 – Fully Convolutional Network pour la segmentation d'image [1]

Architecture U-Net

L'architecture U-Net comme présenté pour la première fois dans l'article [3] est une architecture sous forme de U, une descente et une remontée. Pendant la descente, on va appliquer des successions de filtre de convolution et d'échantillonnage de type max pool 1.3. Dans la figure 1.2 les flèche bleue représente la convolution par des filtres de taille 3x3 et l'application d'une fonction d'activation ReLU sur l'image. Le nombre de canaux correspond au nombre de filtres qu'on a appliqué à l'image, i.e 64 puis 128 ect. Les flèches rouges correspondent à une sous échantillonnage max pool qui va diviser l'image en bloc 2x2 et chacun des blocs seront remplacés par la valeur du pixel maximal (voir 1.3).Cela a pour effet de diviser la taille de l'image par deux, d'ailleurs à chaque application max pool, on multiplie le nombre de filtres par 2 pour capturer plus de structure locale. On réitère le processus jusqu'à atteindre l'étape de remonté. Cette fois si on applique un suréchantillonnage pour chaque flèche verte, cela consiste à augmenter la taille de l'image en rajoutant des zéros, on fait entre guillemet l'application inverse d'un max pooling. Lorsque l'on applique l'up-sampling, on divise le nombre de filtres choisit par 2 pour les nouvelles convolutions. À noter que pour garder des informations de précision spatiale, on applique les convolutions de la remonté à l'image obtenue par up-sampling concaténé avec l'image de la descente qui correspond. Aujourd'hui, on appelle cela des *skip connection* symbolisé par les flèches grise, cela copie une image pour l'envoyer plus loin dans le réseau. Après plusieurs convolutions,ReLu et up-sampling, on atteint la dernière convolution, une convolution 1x1 ou le nombre de filtres permet de choisir le nombre de classes que l'on souhaite pour partitionner l'image.

2

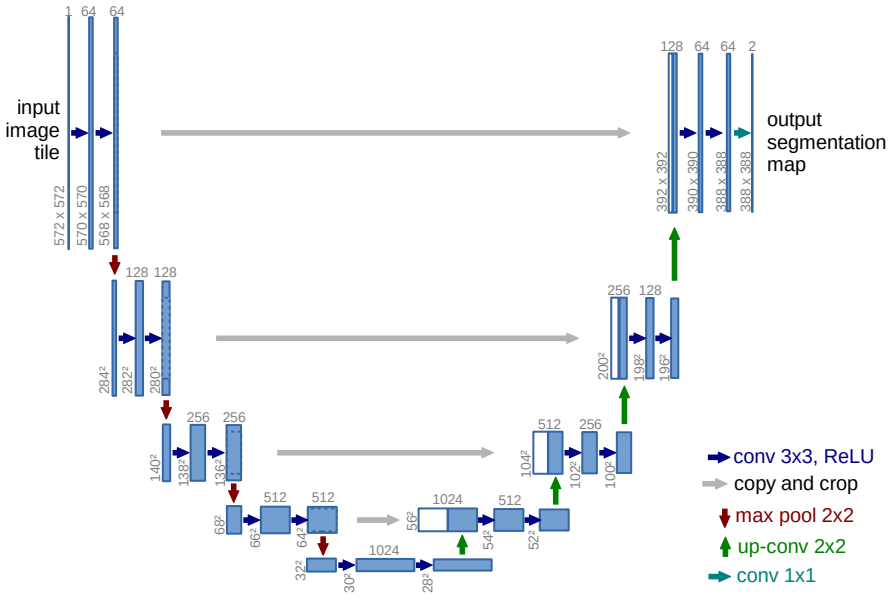

FIGURE 1.2 – Architecture U-Net [3]

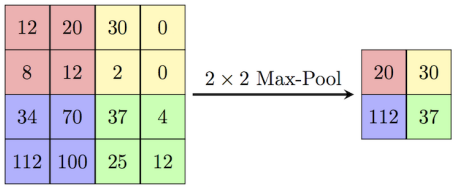

FIGURE 1.3 – Max pooling ://production-media.paperswithcode.com/methods/MaxpoolSample2.png.

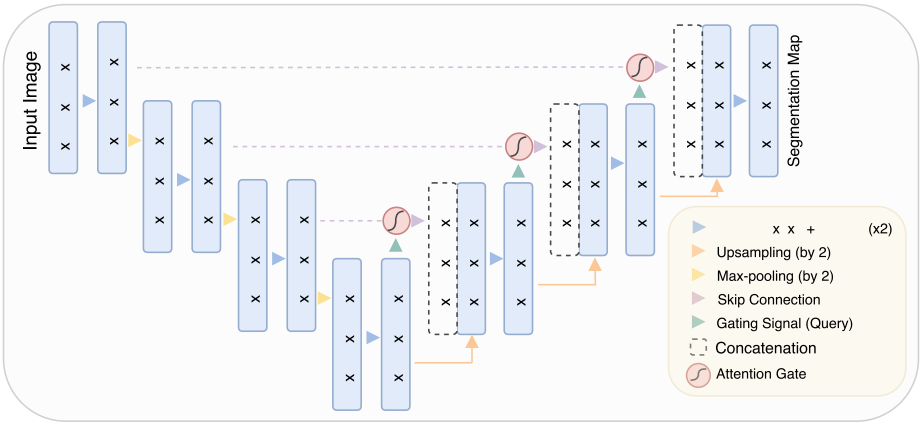

FIGURE 1.4 – Attention U-Net

Remarques

Remarquons une gestion des bords astucieuse consiste à répliquer en miroir l'image du bord pour la fenêtre de contexte qui se retrouverait en dehors de l'image. De plus, l'efficacité du réseau U-Net de l'article [3] réside aussi dans l'ajustement de ses poids. Les poids sont plus fort là où les gradients de l'image sont plus élevés.

Architecture AG U-Net

Dans cette section, nous allons nous concentrer sur la nouveauté apportée par l'article [2] i.e *attention gate* et *gating signal*. Dans la figure, 1.4*l'attention gate* correspond à un filtre d'attention qui ignore les sections de l'image qui ne sont pas pertinentes. Alors que le *gating signal* est une information contextuelle qui sera justement fournie aux *attention gate*. Cela permet de rendre l'information fournie par les *skip connections* plus pertinente pour la segmentation. De plus, cette architecture n'a pas besoin de code externe pour localiser la segmentation d'image. À noter que l'architecture est équivalente à celle du U-Net, outre l'*attention gate* et la convolution 3x3x3 qui correspond à une convolution sur les trois canaux de couleur en plus. Pour améliorer le modèle, on pénalise les gradients appartenant à l'arrière-plan.

Dans la figure 1.5 on représente les réseaux d'*attention gate* données par les équations 1.1 et 1.2. Le coefficient α représente la pondération qu'on applique à chaque pixel de l'image, plus le pixel est pertinent plus la pondération est forte.

$$q_{\text{att}}^l = \Psi^T\big(\sigma_1\big(W_x^T x_i^l + W_g^T g_i + b_g\big)\big) + b_\psi, \tag{1.1}$$

$$\alpha_i^l = \sigma_2\big(q_{\text{att}}^l(x_i^l; g_i; \Theta_{\text{att}})\big), \tag{1.2}$$

où $\sigma_2(x_i, c) = \frac{1}{1+\exp(-x_i,c)}$ correspond à la fonction d'activation

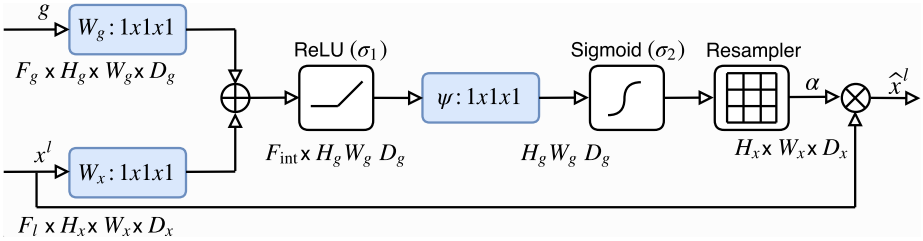

FIGURE 1.5 – Architecture de *l'attention gate*

1.2 Analyse du trade-off précision/contexte

Le but de cette section est de présenter un cas test d'application de réseau U-Net à une base de données que j'ai moi-même créé (avec Chat Gpt), cette base de données est composée d'image, de plusieurs ellipse et rectangle et de leurs labels respectif. On a donc généré 1000 image de taille 128x128 qu'on a partitionné en dataset d'entrainement et de test. Pour évaluer la qualité d'une prédiction, on regarde la métrique IoU "Intersection over Union" qui plus, elle est proche de 1 plus la prédiction chevauche le masque réel. Dans ce cas test, on veut illustrer le compromis entre le contexte et la précision. On va donc faire 3 réseaux U-net qui comporteront plus ou moins de succession de convolution et d'échantillonnage,UNetContext comporte 8 couche en tout (4 descentes et 4 montés), UNetPrecision comporte 4 couche au total et UNetStandard comporte 6 couche. La figure 1.9 représente sur la gauche l'image originale fournit au réseau, au milieu les labels de l'image d'origine et à droite la prédiction fournit par les différents réseaux U-net. On peut remarquer que la segmentation est particulièrement difficile lorsque les formes se chevauchent. Ce pendant le score IoU reste bon pour chacun des réseaux aux alentours de 0.8. Cette expérience ne permet pas de dégager une différence claire de segmentation entre les différents réseaux.

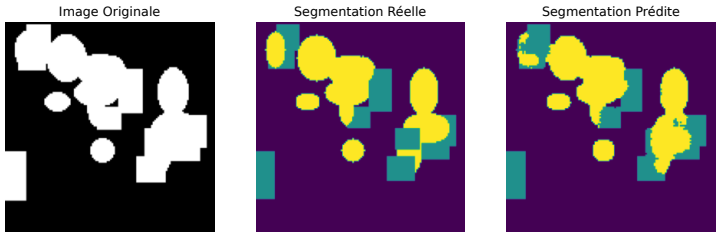

FIGURE 1.6 – Segmentation par UNetContext (score IoU : 0.8190)

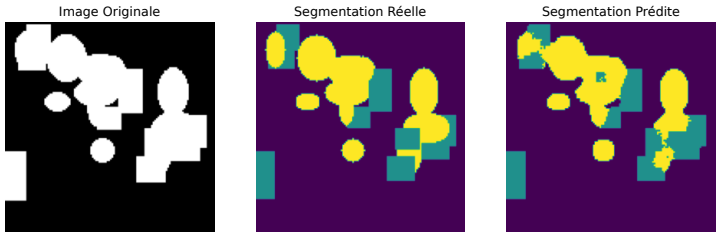

FIGURE 1.7 – Segmentation par UNetPrecision (score IoU : 0.8248)

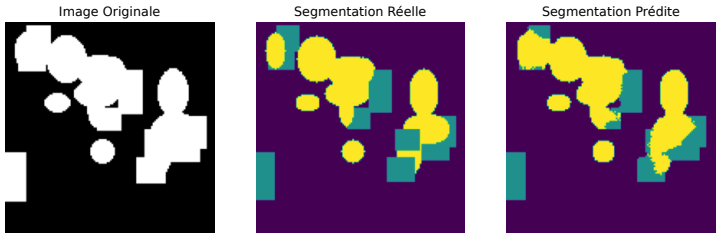

FIGURE 1.8 – Segmentation par UNetStandard (score IoU : 0.7861)

FIGURE 1.9 – Comparaison des résultats de segmentation entre différentes variantes de UNet.

1.2.1 Description des données d'imagerie biomédicale

Dans cette section, nous allons vous présenter les contraintes et les solutions du domaine de l'imagerie biomédicale. En imagerie bio médicale, on a besoin de millier de labels, mais le nombre de données peut être manquant. C'est pourquoi une des méthodes utilisée est l'augmentation de données, elle consiste à appliquer des déformations à nos données et les rajoutés au dataset. À noter que les déformations utilisées peuvent être issues de réels phénomènes physique. Dans l'article [3] ils appliquent des déformations élastiques pour ne pas avoir besoin d'annotation supplémentaire. L'un des effets positifs est de rendre le réseau U-Net robuste à ce type de déformations.

Un autre problème pour la segmentation d'image est la variabilité de taille et de forme des organes comme le pancréas. D'autant plus que l'on souhaite une précision de la classification pour chaque pixel de l'image.

1.3 Conclusion

La lecture de ces articles a été pour moi l'occasion de me familiariser avec les architectures U-Net dans le contexte de l'imagerie biomédicale. J'ai pu aussi mettre au point un code fonctionnel rapidement grâce à l'aide de ChatGPT. Cette étude a permis de mettre en lumière un des points soulevé par l'article [3] i.e la compétition entre la précision et le contexte. Malheureusement, je n'ai pu que montrer partiellement ce compromis, il faudrait d'autre métrique et des exemples mieux choisis pour mieux illustrer ce compromis.

Bibliographie

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[2] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net : Learning where to look for the pancreas, 2018.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation, 2015.