



## **Imagerie Biomédical : Quantification de l'incertitude de la segmentation sémantique de noyaux grâce aux réseaux de neurones bayésiens.**

**Clément Siegrist**

**Ihab Bendidi**

**Université de Paris**

*clement.siegrist@sciencespo.fr*

*ihab.bendidi@etu.u-paris.fr*

### **Abstract**

*Au cours des dernières années, la segmentation des images biomédicales s'est considérablement améliorée, notamment grâce à l'apprentissage profond, à la disponibilité de bibliothèques développées pour tester, standardiser, comparer et affiner les modèles et à la création d'importantes bases de données images. L'architecture d'apprentissage profond la plus réputée pour la segmentation biomédicale est le U-Net, lequel est devenu ces dernières années un gold standard dans l'ensemble de la communauté des chercheurs en vision par ordinateur. Cependant, les approches classiques d'apprentissage profond, que ce soit sur un problème de classification multiclassées ou binaire ne donne qu'une prédiction catégorique pour chaque pixel sans donner d'informations supplémentaires sur le degré de certitude de ces prédictions. Les réseaux neuronaux bayésiens comblent ces lacunes en donnant une mesure de l'incertitude pour chaque prédiction, donnant ainsi des informations statistiques et visuelles supplémentaires au clinicien pour établir un diagnostic plus informé, rigoureux et prudent.*

**Mots-clés :** segmentation sémantique, réseau de neurones probabilistes, réseau de neurones bayésien, Monte Carlo dropout, U-Net, noyaux, cellules.

## 1. Introduction

La segmentation de cellules à une importance particulière dans le domaine de l'analyse biomédicale car elle permet une compréhension plus fine des comportements des cellules dans des situations différentes. La segmentation de cellule permet de mieux comprendre les comportements physiologiques et cellulaires et ainsi d'optimiser le développement de nouveaux médicaments. Les sources et types de cellules étant très différents, les méthodes de détection et segmentations doivent être capables de séparer ces cellules dans des contextes différents, sur des images variées acquises au moyens de divers capteurs. Une des méthodes pour automatiser ce processus est l'utilisation de méthodes d'apprentissage supervisé.

Parmi ces méthodes, l'une de plus en plus utilisée consiste à entraîner un réseau de neurones sur un jeu de données volumineux, qui doit être suffisamment représentatif pour décrire la population cible. Les jeux de données peuvent s'avérer difficiles à acquérir car ils doivent garantir une certaine exhaustivité des caractéristiques des images et lesquelles nécessitent de surcroît d'être labellisées. De plus, les techniques de régularisation peuvent ne pas être suffisantes pour généraliser la performance du réseau sur l'ensemble des scénarios possibles. Ce défaut peut être adressé en introduisant des estimations de certitude, qui permettent d'interpréter les résultats, et d'identifier les zones où les résultats sont ambigus.

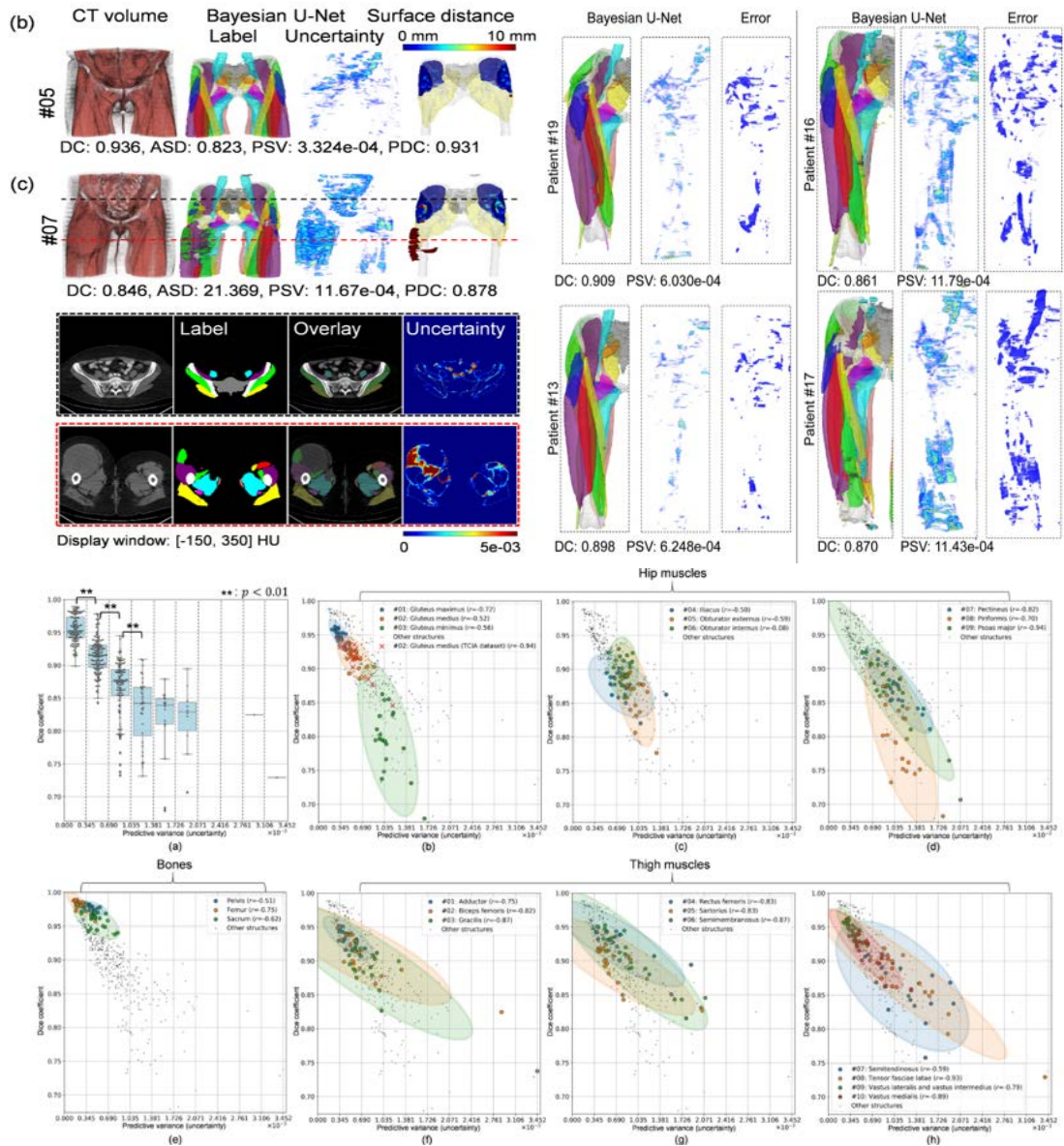
Pour ce faire, nous avons modifié l'architecture standard du U-Net pour réaliser une inférence bayésienne. Le U-Net est une architecture de réseaux de neurones à convolutions de type autoencodeur encore abondamment utilisées et déclinées en plusieurs sous-types d'architectures. Nous effectuons ensuite une évaluation extensive de notre modèle en utilisant une base de données de cellules de Kaggle, tout en le comparant aux résultats d'un U-Net standard entraîné sur les mêmes données.

Bien que notre approche modifie, ou du moins ajoute certaines opérations mathématiques à l'apprentissage profond déterministe, le U-Net reste la colonne vertébrale de notre architecture convolutionnelle. Le U-Net reste à ce jour et est toujours un gold standard dans la communauté de la segmentation d'images biomédicale. A ce propos, voir le benchmark récent de L. Putzu et Fumera (2020) ainsi que l'étude de Le Hou et al [2] basée sur un jeu de données d'environ 5 milliards de noyaux provenant du *Cancer Genome Atlas* [12]. De plus, le lecteur souhaitant trouver un aperçu de l'état de l'art des techniques d'apprentissage profond déterministes appliquées à la segmentation de noyaux peut se référer à l'étude des résultats des travaux présentés dans le cadre du Défi de segmentation et de classification des noyaux multi-canaux 2020 (MonuSac 2020) [3] ainsi que le document résumant les résultats du Data Science Bowl 2018 qui portait sur la segmentation du noyau de cellules[4]. Les githubs des meilleurs challengers de cette compétition sont disponibles et listés sous les références de la publication [4].

## 2. Travaux similaires

### 2. 1. Segmentation d'images biomédicales par réseaux de neurones probabilistes

Dans [6], les auteurs de l'étude ont modifié un U-Net standard afin de procéder à de l'inférence bayésienne grâce à de l'échantillonnage de Monte-Carlo pour de la segmentation des couches des photorécepteurs rétiniens à partir d'images OCT. J. Senapti et al [7] ont testé et comparé quatre architectures probabilistes différentes sur des images de cerveaux issues de scans IRM. D'autres chercheurs ont utilisé les mêmes modèles pour de la segmentation multi classes d'images CT de muscles [13]. Un aperçu de leurs résultats ainsi que les corrélations négatives entre la précision et l'incertitude sont représentés sur la **Figure 1**.



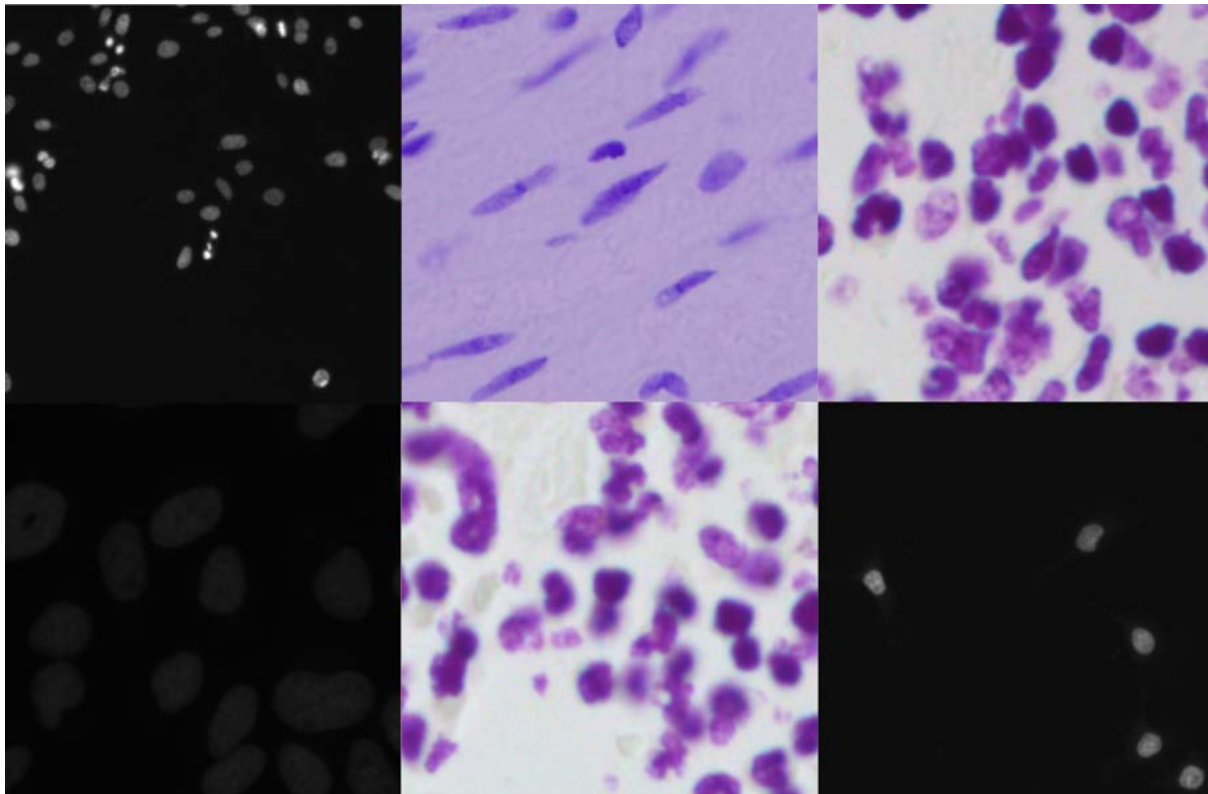
**Figure 1.** Résultats visuels de la segmentation et de l'estimation de l'incertitude et de l'erreur, corrélations entre variance prédite et coefficient de Dice.

## 2. 2. Segmentation de noyaux avec des approches floues (fuzzy logic)

D'autres approches mathématiques visant à donner une représentation probabiliste, ou visant à donner un degré d'incertitude ou d'appartenance des données à une ou plusieurs classes telles que les approches floues ont aussi été utilisées récemment et appliquées à la segmentation d'images biomédicales. Des travaux récents ont utilisé des réseaux neuronaux flous pour quantifier l'incertitude des prédictions dans le domaine de la segmentation sémantique de sur des images issues d'échographies du sein [5]. D'autres travaux ont utilisé un perceptron multi-couches flous avec de l'analyse fractale en post-traitement pour la segmentation et la caractérisation des lésions cutanées [8]. Les auteurs affirment obtenir de meilleurs résultats qu'avec les réseaux neuronaux entièrement convolutionnels. Des algorithmes de clustering plus légers comme Fuzzy-C Means peuvent également être utilisés pour la segmentation sémantique avec visualisation de l'incertitude.

## 3. Jeu de données utilisé

Nous avons créé un jeu de données personnalisé se basant sur le jeu d'entraînements composé d'images et de labels fourni par Kaggle dans le cadre du 2018 *Data Science Bowl*. Ce jeu de données est composé d'images de nucleus (noyau de cellules). Le challenge vise à développer un algorithme capable de facilement détecter et segmenter les noyaux des cellules afin de les comptabiliser, de repérer et distinguer les cellules cancéreuses des cellules saines, d'identifier et de traquer leurs déplacements afin d'observer la réaction d'un patient à un traitement au niveau moléculaire. Les images ont été acquises par différents instruments et dans différentes conditions d'illumination, de magnification et d'autres modalités (fluorescence). Le jeu de données était divisé en trois parties : un jeu d'entraînement de 670 images avec les labels fournis et deux autres jeux de tests sans labels. Nous avons donc téléchargé le premier jeu et divisé le jeu de données en jeu d'entraînement, de validation et de test pour le modèle bayésien ainsi que le U-Net standard. Les données images sont en format RGB tandis que les labels sont binaires (0, 1). Nous allons à présent commenter certaines des images de notre jeu de données visibles sur la **Figure 2**.



**Figure 2.** Exemples d'images de notre jeu de données.

On observe une importante diversité des caractéristiques visuelles des images de cellules ainsi que les arrières plans des images. Il est donc nécessaire d'entraîner un modèle capable de généraliser sur plusieurs types de cellules. La **Figure 3** ci-dessous présente le format des labels existants dans la base de données : pour chaque image contenant plusieurs cellules, sont attachées plusieurs images, chacune contenant les labels binaires de la localisation d'une seule cellule dans l'image.



**Figure 3.** Format des labels de notre jeu de données.

Une première étape a été de regrouper les nombreux labels liés à une seule image de cellules en un seul label. En utilisant OpenCV, nous avons donc pu superposer les nombreuses images de labels dans une seule image. La **Figure 4** montre le format des données après avoir effectué cette transformation.

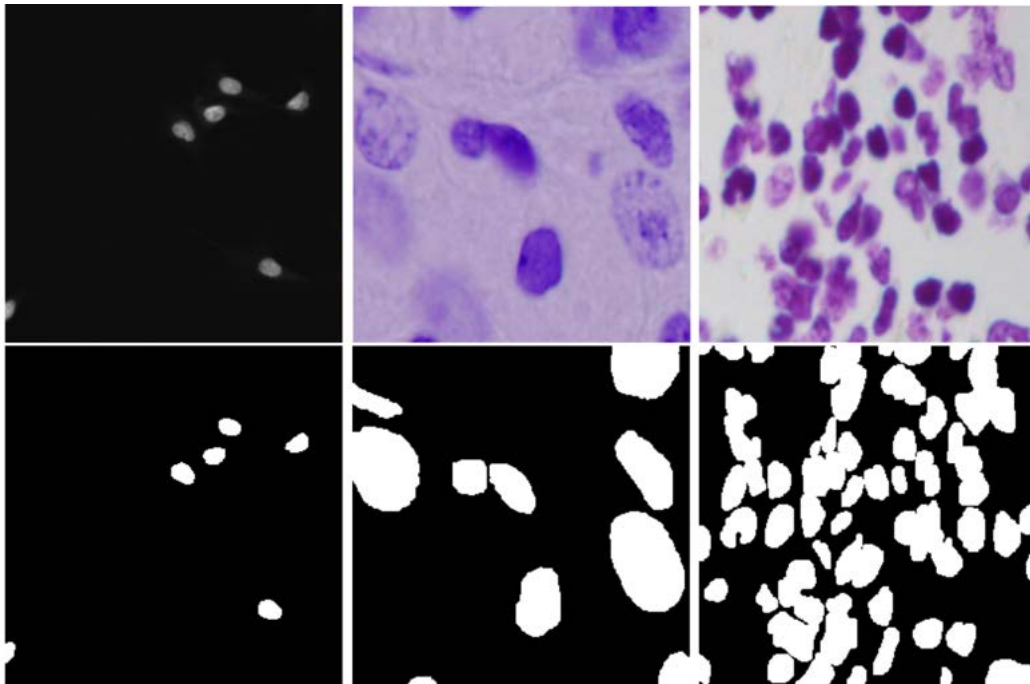


Figure 4. La première ligne correspond aux images originales de cellules, la seconde à leurs labels.

Nous avons ensuite normalisé les images et modifié leur taille. Enfin, étant donné que les valeurs des pixels des labels sont entre 0 et 255, nous avons binarisé toutes les valeurs en leur attribuant soit 0 soit 1. Le jeu de données a été ensuite séparé en données d'entraînement (70%), de validation (10%) et de test (20%).

#### 4. Entraînements et résultats de l'architecture profonde probabiliste et de l'architecture profonde classique

Après description des architectures utilisées, de leurs principes de fonctionnement et de leurs caractéristiques mathématiques, nous détaillerons notre processus d'entraînement des modèles et commenterons les résultats obtenus.

##### 4. 1. Architectures Probabilistes

Les deux principales architectures de quantification de l'incertitude, de *Uncertainty Quantification* en anglais [9], sont les réseaux de neurones bayésiens et les réseaux de neurones à Dropout de type Monte-Carlo. Pour résumer les différences entre les deux types d'architectures de façon synthétique, la première génère une estimation de l'incertitude en utilisant comme distributions à priori l'ensemble des poids  $\mathbf{W}$  tandis que la seconde génère de l'incertitude au niveau de chaque couche de Dropout.



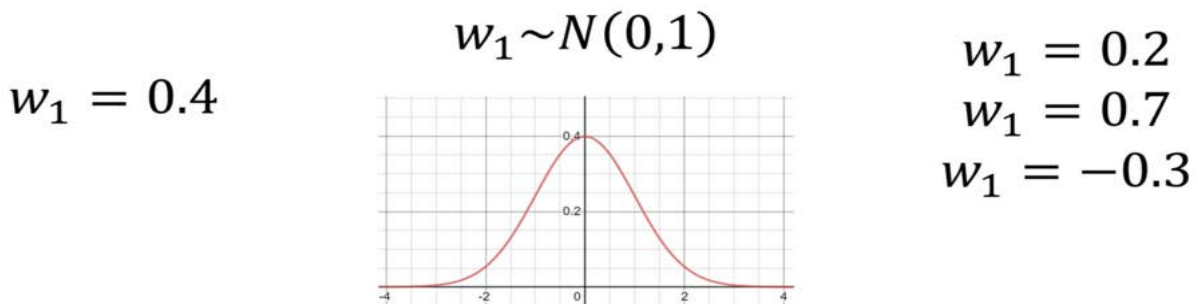
### 4. 1. 1. Réseaux de neurones profonds bayésien

Le deep learning bayésien permet de calculer les incertitudes épistémiques en modélisant une distribution à priori  $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$  sur les poids  $\mathbf{W}$  du réseau, pour un ensemble d'images d'entraînement donné  $\mathbf{X} = \{X_k\}$  et les labels  $\mathbf{Y} = \{Y_k\}$ ,  $k = 1, \dots, N$  correspondant. En pratique, trouver la distribution à priori exacte est infaisable compte tenu du nombre très élevé de paramètres dans une réseau de neurone, mais on peut obtenir une approximation  $q(\mathbf{W})$  en utilisant l'inférence variationnelle (obtention par approximation de la distribution à priori de la loi de bayes), en minimisant la divergence de Kullback-Leibler (KL) [14]  $KL(q(\mathbf{W}) \parallel p(\mathbf{W}|\mathbf{X}, \mathbf{Y}))$ . Dit autrement, Si la plupart des réseaux neuronaux apprennent des estimations ponctuelles de leurs poids, ces estimations ne représentent pas entièrement l'incertitude inhérente aux valeurs des poids. Au lieu de cela, dans un RCPB<sup>1</sup>, les pondérations sont implicitement décrites comme des distributions de probabilité (multivariées). La **Figure 5** donne une explication simplifiée de la procédure d'estimation des poids utilisée.

Dans un réseau de neurones classiques chaque neurones apprend une valeur unique de son poids, sélectionné selon une loi de probabilité choisie par l'utilisateur (he\_normal, gloriot\_uniform)

Un réseau de neurones bayésiens représente les poids comme une distribution de probabilité. Les valeurs centrales (autour de zéro) ont plus de chances d'être échantillonnées.

Pendant la propagation avant on échantillonne en fonction de la distribution de probabilité pour obtenir plusieurs estimations des poids.

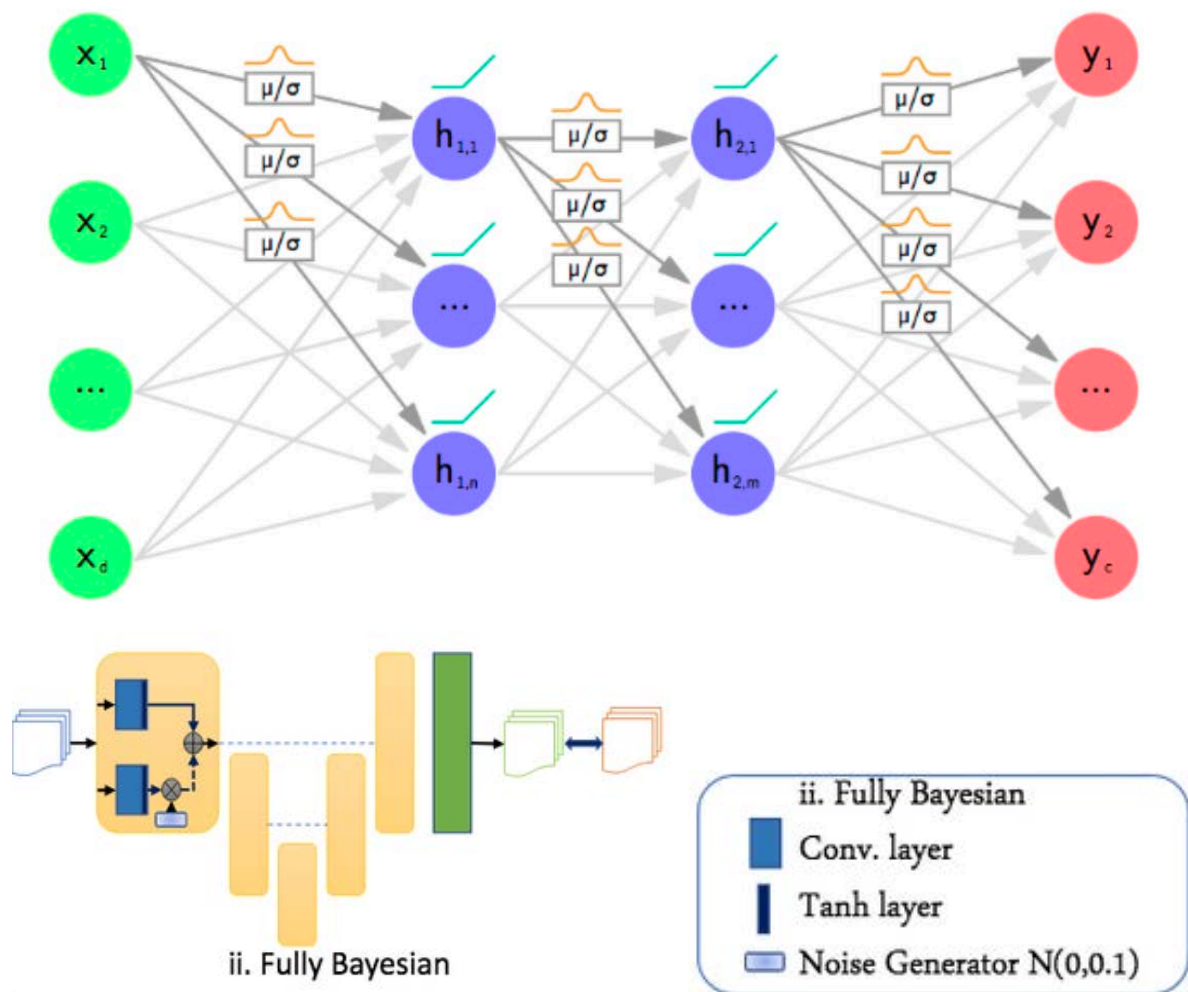


**Figure 5.** Description de la détermination des poids dans un réseau de neurones bayésien.

Enfin, la descente du gradient se fait différemment qu'avec les réseaux de neurones classiques. La *backpropagation through times* n'est pas possible dans le cas des réseaux probabilistes car il n'est pas possible de calculer le gradient d'une variable échantillonnée. Toutefois, Kingma et [14]. ont montré qu'il est possible de paramétrer la variable aléatoire comme une variable déterministe pour le calcul de la *backpropagation*. Par exemple, prélevons, un poids  $w$  à partir d'une distribution normale avec la moyenne  $\mu$  et la variance  $\sigma^2$ . Une reparamétrisation valide de cette loi précédemment définie est donnée comme suit :  $w = \mu + \sigma\epsilon$  où  $\epsilon$  est une variable de bruit auxiliaire échantillonnée à partir d'une distribution normale standard. Ainsi, au lieu d'une opération d'échantillonnage, nous avons une combinaison affine, qui est facilement utilisée dans la rétropropagation. Des calculs plus complexes sont nécessaires pour optimiser ce calcul est l'adapter à l'architecture bayésienne. Cette méthode se nomme technique locale de paramétrisation. Pour plus de

<sup>1</sup> Réseau de Neurones Convolutionnels Profond Bayésien.

détails consulter l'étude de Kingma [14]. La **Figure 6** correspond à une description schématique de l'architecture d'un réseau de neurones bayésien.



**Figure 6.** Architecture simplifiée d'un réseau de neurones totalement bayésien.

#### 4. 1. 2. Réseaux de neurones profonds avec Dropout de Monte-Carlo

Nous allons décrire un peu plus en détail les principes mathématiques de cette méthode puisque c'est l'architecture probabiliste que nous avons utilisé pour la segmentation de notre jeu de données.

Dans un réseau de neurones classique, déterministe, le Dropout va supprimer les poids avec une probabilité fixe déterminée par l'expérimentateur (par exemple 0.2, 0.4, 0.6). De surcroît, le Dropout n'est effectué que pendant la phase d'entraînement afin d'empêcher le sur apprentissage. Pendant la phase de test, tous les neurones, correctement entraînés grâce au Dropout sont utilisés. Le Dropout de Monte-Carlo quant à lui va effectuer plusieurs Dropout pour chaque couche de Dropout avant chaque nouvelle couche à convolution ou dense [7] pendant l'entraînement et l'inférence.



Ce faisant nous obtenons plusieurs “*distributions*” des valeurs des poids avant chaque couche neuronale et/ou plusieurs distributions des prédictions après les fonctions d’activation de chaque couche et la softmax de la couche dense. Afin que le lecteur puisse avoir une intuition du fonctionnement mathématique du Dropout de Monte Carlo, une description schématique de l’architecture est représentée dans la **Figure 7** et un exemple d’échantillonnage de Monte-Carlo classique ainsi que les résultats des différentes distributions sont représentés dans la **Figure 8**.

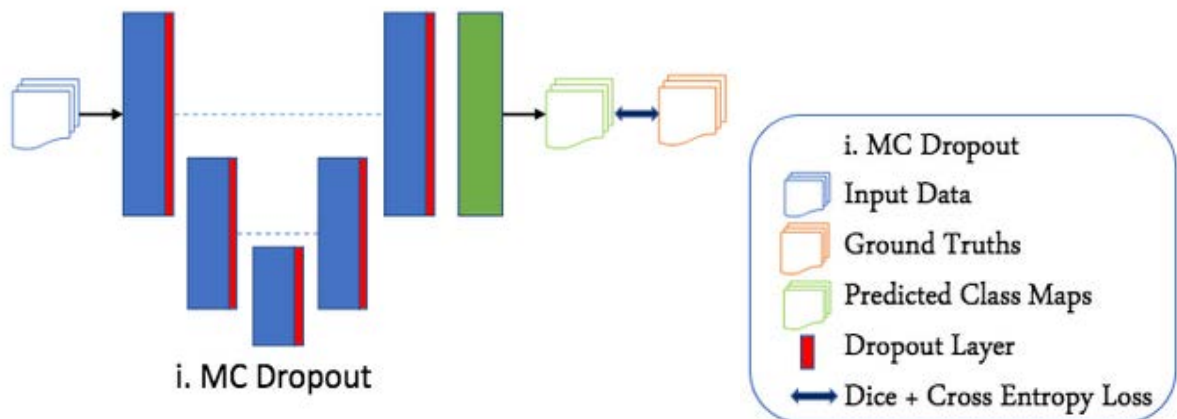


Figure 7. Architecture simplifiée d'un Unet avec Dropout de Monte Carlo.

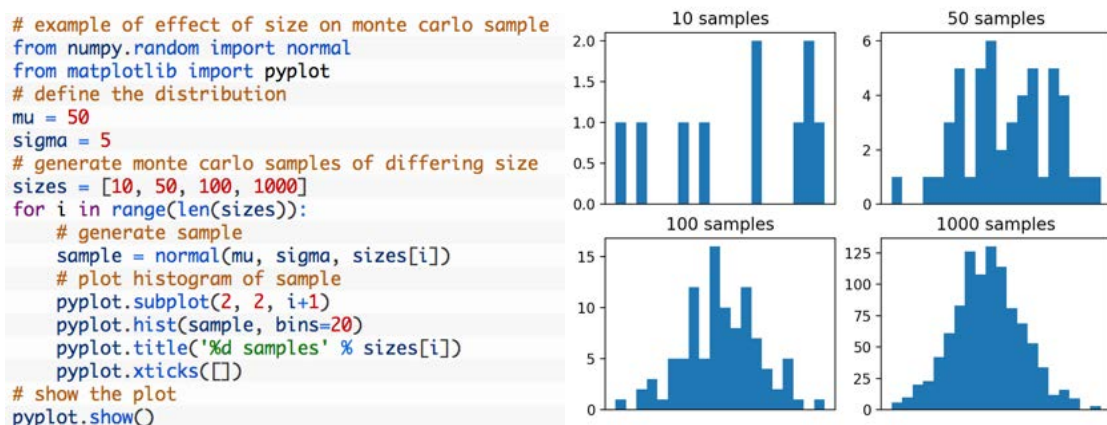


Figure 8. Code python permettant d'effectuer une simulation de Monte Carlo et d'en visualiser la représentation graphique.

Nous allons décrire plus précisément comment fonctionne ce type d'architecture dans les paragraphes suivants.

Supposons que nous avons un jeu d'entraînement composé d'images  $\mathbf{X} = \{x_1, \dots, x_n\}$  et de ses labels  $\mathbf{Y} = \{y_1, \dots, y_n\}$ . On considère le label prédit  $y^*$  d'une image encore inconnue  $x^*$ . Un réseau de neurones classique, “*déterministe*” représentera la probabilité de ce label  $y^*$  sachant l'image  $x^*$  comme suit :  $p(y^*|x^*) = \text{Softmax}(f(x^*; \mathbf{W}))$ . Dans les réseaux de neurones “*probabilistes*” ou “*bayésiens*”, la probabilité du label  $y^*$  est donnée par la marginalisation, c'est à dire dans notre cas l'addition des probabilités conditionnelles pour chaque échantillonnage des poids  $\mathbf{W}$  du réseau comme suit :

$$p(y^* = c | \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(y^* = c | \mathbf{x}^*, \mathbf{W}) p(\mathbf{W} | \mathbf{X}, \mathbf{Y}) d\mathbf{W} \quad (1)$$

où  $y^* \in \mathcal{Y}^*$  est le label prédit du pixel,  $c$  est la classe du label, et  $p(\mathbf{W} | \mathbf{X}, \mathbf{Y})$  est la distribution à posteriori. Gal et al. ont démontré dans [10] que la distribution à priori est équivalente à une distribution de dropout masquée  $q(\hat{\mathbf{W}}) = \mathbf{W} \cdot \text{diag}(\mathbf{z})$  et  $\mathbf{z} \sim \text{Bernoulli}(\theta)$ , avec  $\theta$  étant le ratio de Dropout. Alors l'Équation. (1) peut-être approximée en minimisant la divergence de Kullback-Leibler  $KL(q(\hat{\mathbf{W}}) || p(\hat{\mathbf{W}} | \mathbf{X}, \mathbf{Y}))^2$  comme suit :

$$\begin{aligned} p(y^* = c | \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) &\approx \int p(y^* = c | \mathbf{x}^*, \hat{\mathbf{W}}) q(\hat{\mathbf{W}}) d\hat{\mathbf{W}} \\ &\approx \frac{1}{T} \sum_{t=1}^T \text{Softmax}(\mathbf{f}(\mathbf{x}^*, \hat{\mathbf{W}}_t)). \end{aligned} \quad (3)$$

où  $T$  correspond au nombre d'échantillonnages de Monte-Carlo. Cette estimation de Monte-Carlo correspond au Monte-Carlo Dropout précédemment décrit [7].

Enfin, une mesure d'incertitude possible pour évaluer les qualités des prédictions est la variance prédictive est définie comme suit :

$$\begin{aligned} \text{Var}(y^* = c | \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) &\approx \frac{1}{T} \sum_{t=1}^T \text{Softmax}(\mathbf{f}(\mathbf{x}^*, \hat{\mathbf{W}}_t))^T \text{Softmax}(\mathbf{f}(\mathbf{x}^*, \hat{\mathbf{W}}_t)) \\ &\quad - p(y^* | \mathbf{x}^*, \mathbf{X}, \mathbf{Y})^T p(y^* | \mathbf{x}^*, \mathbf{X}, \mathbf{Y}). \end{aligned} \quad (4)$$

## 4. 2. U-Net classique

Le U-Net [15] est un réseau de neurones convolutifs pyramidal de type autoencoder, c'est à dire que le nombre de filtres de convolutions par couches diminue puis augmente jusqu'à atteindre le nombre de départ. De surcroît les couches sont partiellement interconnectées et non totalement séquentielles c'est-à-dire que chaque couche n'est pas uniquement connectée avec la couche précédente et la couche suivante. Une description schématique du U-Net est représentée dans la **Figure 9**. L'encodeur se compose de 5 blocs convolutifs, avec 16, 32, 64, 128, 256, 512, 1024 filtres par blocs suivis de 2 max-pooling et avec une couche de BatchNormalisation. Ces blocs utilisent deux couches 2\*2 convolutionnelles, chacune d'entre elles suivie d'une BatchNormalisation et d'une couche avec la fonction d'activation ReLU [15]. Le décodeur se compose de 4 blocs d'upsampling avec 1024, 512, 256, 128 et 64, 32, 16 canaux de sortie. La couche finale est une couche dense composée de 2 neurones correspondant au niveau confiance de la prédiction pour les deux classes (cellules, background).

---

<sup>2</sup> Détaille dans l'appendice 1. de la divergence de Kullback-Leibler.

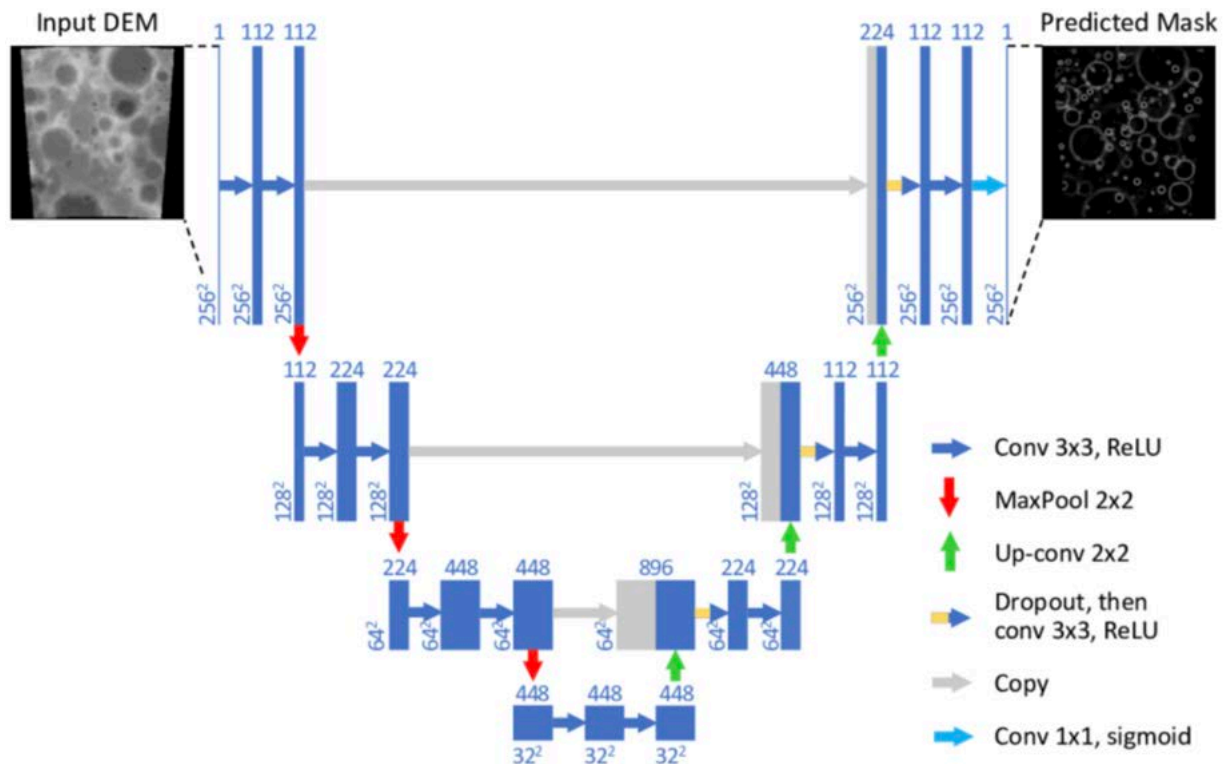


Figure 9. Schéma de l'architecture d'un U-Net classique, déterministe.

### 4. 3. Architectures et modèles utilisés

Nous avons utilisé deux architectures dans le cadre de notre étude, un U-Net classique et un U-Net Probabiliste avec Dropout de Monte-Carlo.

#### 4. 3. 1. Réseau à Dropout de Monte-Carlo

L'architecture probabiliste utilisée dans notre travail est un réseau de neurones à Dropout de type Monte-Carlo s'inspirant des publications [13][6] dont le code est disponible sur [github](https://github.com). Dans les deux publications, les auteurs ont surnommé les méthodes utilisées "*Bayesian Unet*" [13][6] par convenance, mais nous tenons à préciser que, comme décrit précédemment, un réseau de neurones purement bayésien diffère d'un réseau de neurones à Dropout de Monte-Carlo. Dans les cas des études citées et du modèle que nous avons utilisé, il s'agit bien d'un modèle avec Dropout de Monte-Carlo.

#### 4. 3. 2. Réseau U-Net

Nous avons utilisé une architecture de U-Net similaire à celle décrite dans la sous partie 2 précédente, à la seule différence étant que le nombre de la taille maximale d'un bloc convolutif choisie est de 256 afin accélérer l'entraînement.

## 5. Résultats

Le modèle U-Net standard a été entraîné sur 10 époques. Nous obtenons un score de précision de 100%. Les images ont été redimensionnées à une taille de 256x256, et un *batchsize* de 4 a été utilisé, avec une fonction de coût d'entropie croisée, Adam comme optimiseur, et un pas d'apprentissage de 0.0001. Les **Figures 10, 11** ci-dessous montrent respectivement l'évolution de l'entraînement, ainsi que des exemples de segmentation réalisés par le modèle sur les données de test.

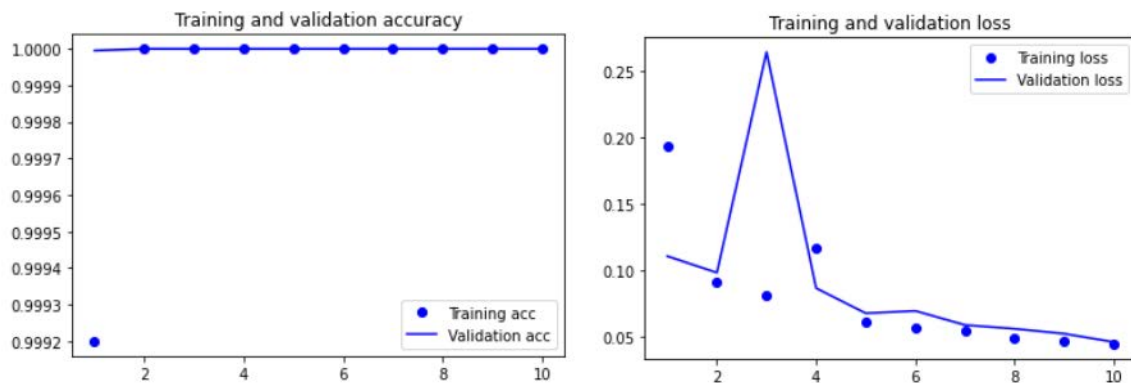


Figure 10. Évolution des mesures de performances lors de l'entraînement du U-Net standard

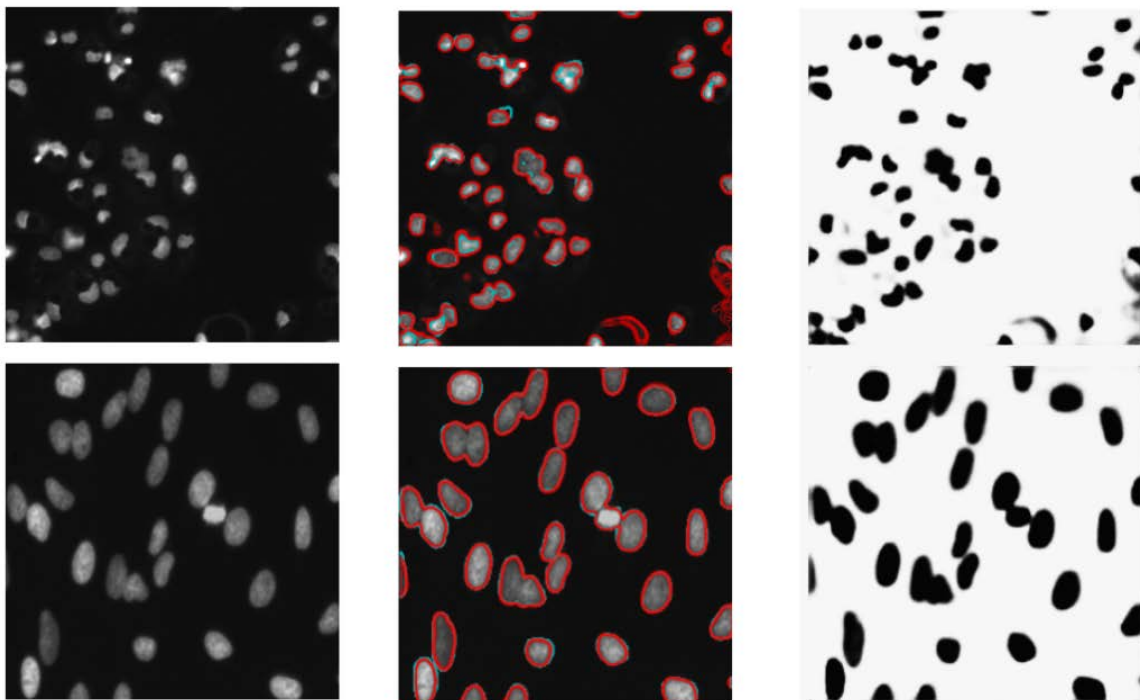
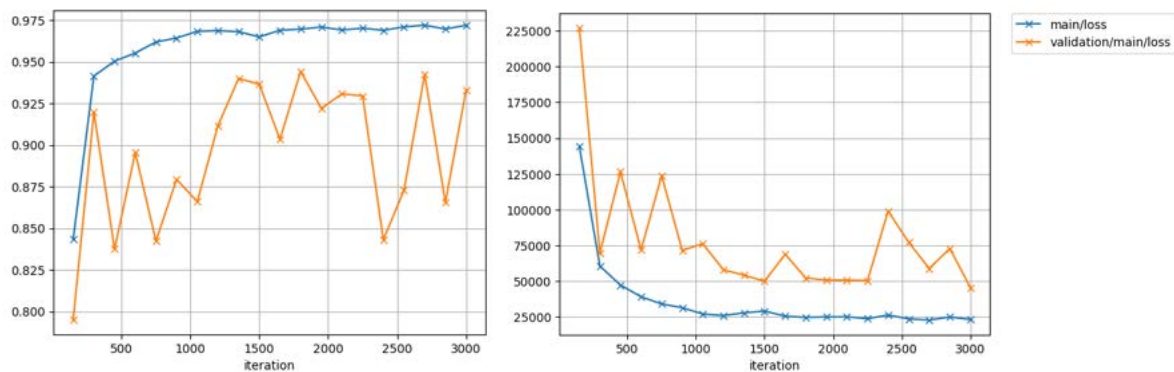


Figure 11. Résultats du U-Net standard sur les données de test

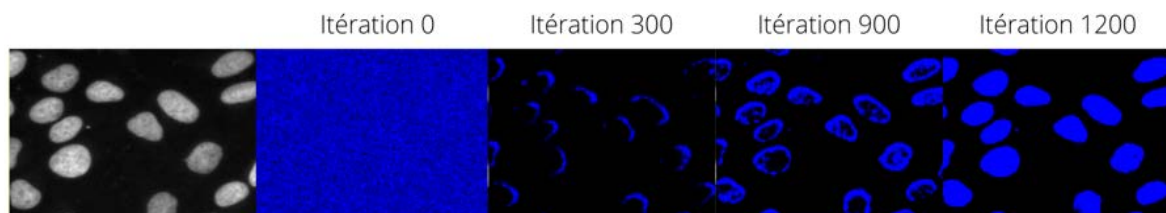
Pourtant, malgré les performances notables de ce modèle sur ce jeu de données, plusieurs situations seront très différentes des données lors de l'utilisation du modèle dans la vie réelle. Le manque d'interprétabilité représente un défaut conséquent des réseaux de neurones. Ainsi, beaucoup restent encore réticent à l'idée de confier à un modèle la

responsabilité d'une prise de décision sans supervision humaine, car ils commettent encore des erreurs grossières difficiles à expliquer à fortiori. L'introduction de mesures d'incertitudes dans les réseaux de neurones bayésiens permet de déterminer les situations dans lesquelles le modèle n'est pas certain de sa prédiction. Notre modèle bayésien a été entraîné sur 3000 itérations, avec une taille de batch de 16, l'entropie croisée comme fonction de coût, et Adam comme optimiseur. La **Figure 12** montre l'évolution des mesures de performances de notre modèle bayésien lors de l'entraînement. La précision atteint 94% sur les données de validation.



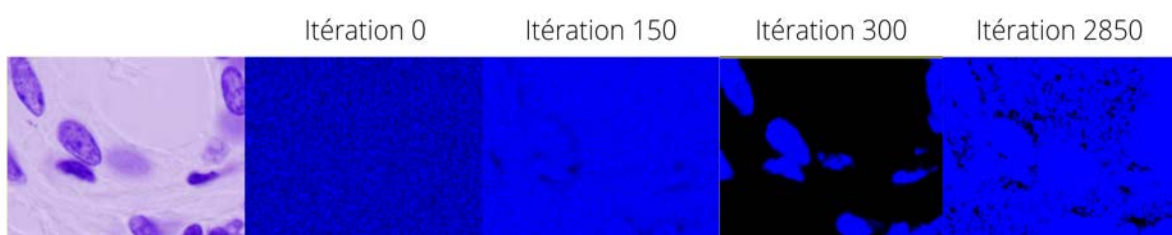
**Figure 12.** Évolution des courbes de loss et accuracy pour le modèle bayésien.

Nous avons aussi pu effectuer un suivi des prédictions de notre modèle bayésien au fil des itérations, comme le montre la **Figure 13**. Lors de la toute première itération la segmentation échoue. Cependant dès la 300ème itération les cellules commencent à être distinguées et nous obtenons une segmentation quasi-parfait à la 1200ème itération.



**Figure 13.** Évolution des segmentations sur les données de validation des prédictions du modèle bayésien au fil de l'entraînement.

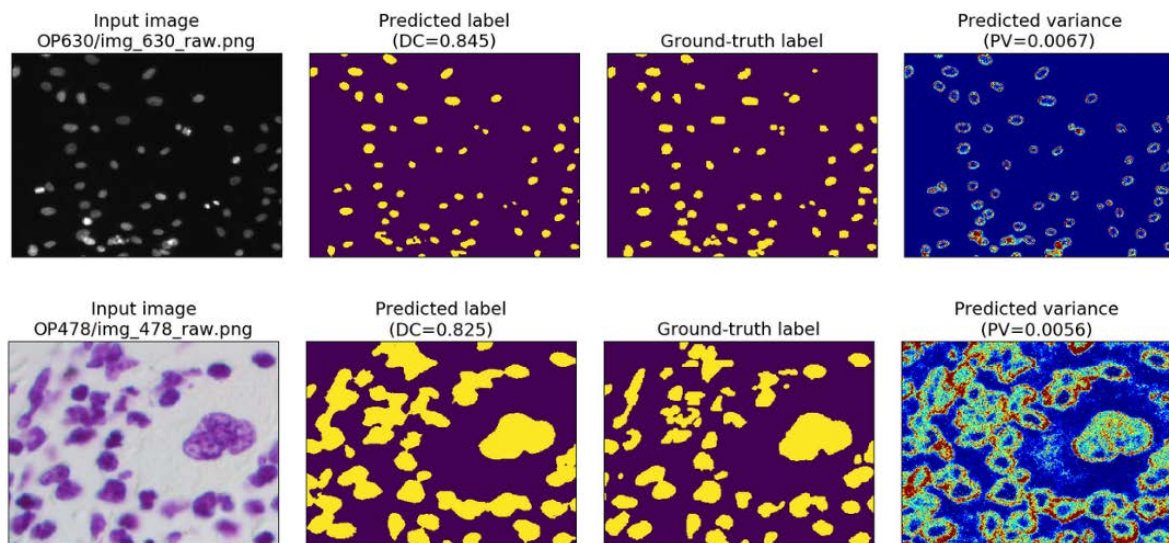
Pourtant, la figure ci-dessous montre aussi que le modèle peut parfois “oublier” des caractéristiques apprises au fil des itérations. Comme on peut l'observer sur la **Figure 14**, les segmentations sont de meilleures qualités à la 300ème itération qu'à la 2850ème. Ce phénomène est appelé « *Catastrophic Inference* » ou « *Catastrophic Forgetting* ».



**Figure 14.** Oubli catastrophique du modèle à un certain nombre d'époques.

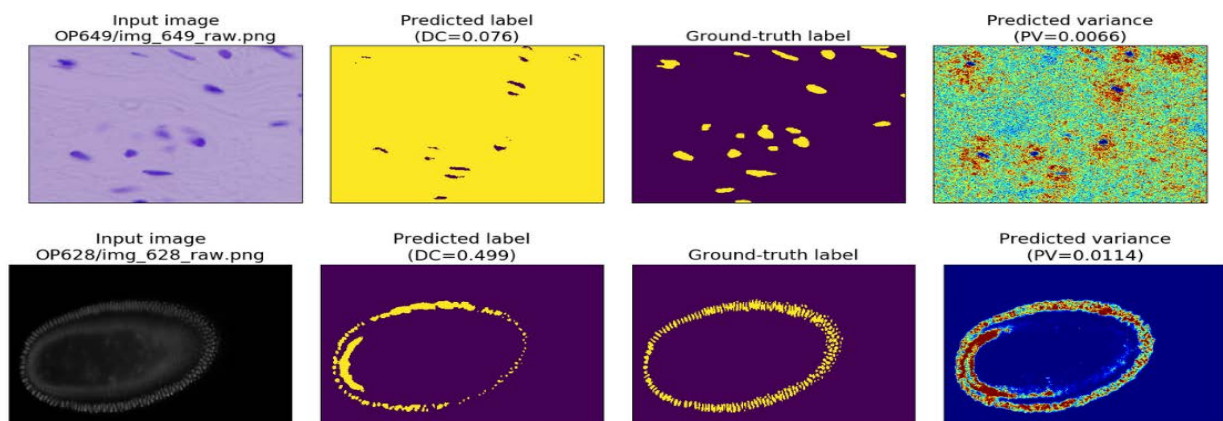


Les résultats du modèle témoignent de la pertinence de l'approche bayésienne. La **Figure 15** représente les segmentations réalisées sur les données de test. La métrique « *Predictive Dice* », qui permet de quantifier l'incertitude du modèle et d'évaluer la robustesse des prédictions. La « *Predictive Dice* » correspond à un score F1 mais adaptée au modèle probabiliste. Le modèle est plutôt certain quant à ses prédictions et à raison comme on peut l'observer sur les images de la **Figure 15**. Le modèle utilise aussi la métrique « *Predictive Variance* » pour visualiser le degré d'incertitude pour chaque pixel grâce à une carte de chaleur ou « *heatmap* ». Plus la couleur est chaude et tend vers le rouge, plus le modèle est incertain et plus la couleur est et tend vers le bleu, plus le modèle est confiant dans ses prédictions.



**Figure 15.** Prédictions et résultats positifs sur les données de test. Les deux métriques utilisées sont le coefficient de Dice prédictif (*Predictive Dice Coefficient*) et la variance prédictive (*Predictive Variance*).

Sur la dernière image de la première ligne **Figure 16**, on observe que l'incertitude est logiquement répartie autour des cellules. De même pour la dernière image de la dernière ligne, la zone mal prédite à l'intérieur du noyau a une variance élevée. Cela indique que le modèle peut être potentiellement facilement amélioré, à l'entraînement ou par du pré ou post-processing de l'image.



**Figure 16.** Les prédictions et résultats faux ont un très fort score d'incertitude (*predicted variance*).



## 7. Limites

Notre travail présente plusieurs limites. Premièrement nous n'avons pas effectué une comparaison rigoureuse des performances des deux modèles utilisés. En effet, ces derniers n'ont pas été entraînés avec le même nombre d'époques, avec les mêmes hyperparamètres, le même learning rate et les métriques utilisées pour quantifier la précision et la qualité des segmentations sont parfois différentes. De surcroît, nous n'avons pas encore effectué une comparaison rigoureuse des résultats des prédictions pour un ensemble d'images statistiquement significatif, ni utilisé de matrices de confusions ou d'autres métriques d'évaluation des performances comme la courbe AUC. De plus, nous n'avons testé qu'une seule architecture probabiliste et qu'une seule architecture classique. D'autres extensions ont été créées pour les architectures classiques qui auraient pu être testées afin d'évaluer leur performance et comparer les résultats obtenus à ceux des modèles probabilistes. De surcroît, nous n'avons pas effectué de comparaison portant sur le temps d'entraînement. Il serait nécessaire de tester la performance de nos modèles sur d'autres types de données biomédicales afin d'évaluer la capacité de ces architectures à généraliser.

Aussi, d'autres approches mathématiques, telles que les réseaux de neurones flous ou les algorithmes de clustering flous devront être testés et comparés aux approches utilisées dans cette étude.

## 8. Conclusion

Nous avons démontré qu'un réseau de neurones probabiliste avec Dropout de Monte-Carlo parvenait à des résultats visuels intéressants sur un problème de segmentation de noyaux de cellules. L'apport informatif de notre approche peut s'avérer utile pour le clinicien et surpasse les modèles traditionnels déterministes en termes d'exhaustivité de l'information fournie. De surcroît, notre méthode est d'autant plus prometteuse que le réseau a été entraîné pendant un nombre d'époque assez bas, sur un jeu de données relativement petit et sans réelle optimisation des hyper-paramètres. Nous pouvons donc conclure que l'apprentissage profond probabiliste représente une voie prometteuse pour la segmentation d'images biomédicales.

## 9. Bibliographie

[1] Lorenzo Putzu and Giorgio Fumera (2020) *An Empirical Evaluation of Nuclei Segmentation from H&E Images in a Real Application Scenario*, Applied Sciences MDPI.

[2] Le Hou et al (2020) *Dataset of segmented nuclei in hematoxylin and eosin stained histopathology images of ten cancer types*, Nature. Code available on github.  
<https://www.nature.com/articles/s41597-020-0528-1>

[3] <https://monusac-2020.grand-challenge.org/>

- [4] Juan. C. Ceicedo et al (2019) *Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl*, Nature methods.
- [5] K. Huang, Y. Zhang et al (2019) *Fuzzy Semantic Segmentation of Breast Ultrasound Image with Breast Anatomy Constraints*, arXiv.
- [6] J. I. Orlando et al (2019) *U2-Net: A Bayesian Model with epistemic uncertainty feedback for photoreceptor layer segmentation in pathological OCT Scans*, arXiv.
- [7] J. Senaptj et al (2020) *Bayesian Neural Network for Uncertainty Estimation of Imaging Biomarkers*, arXiv.
- [8] Abder-Rahman Ali et al (2020) *A Novel Fuzzy Multilayer Perceptron for the Detection of Irregularity in Skin Lesion Border using Dermoscopic Images*, Front. Med., 07 July 2020.
- [9] T. LaBonte et al (2020) *We know Where we Don't Know: 3D Bayesian CNN for Credible Geometric Uncertainty*, arXiv.
- [10] Yarin Gal, Zoubin Ghahramani (2016), *Dropout Bayesian Estimation: Representing Uncertainty in Deep Learning*, Proceedings of The 33rd International Conference on Machine Learning.
- [11] J. Senaptj et al (2020) *Bayesian Neural Network for Uncertainty Estimation of Imaging Biomarkers*, arXiv
- [12] ["The Cancer Genome Atlas homepage"](#). [NCI](#) and the [NHGRI](#).
- [13] Y. Hiasa et al (2019) *Automated Muscle Segmentation from Clinical CT using Bayesian U-Net for Personalized Musculoskeletal Modeling*, arXiv.
- [14] Diederik P. Kingma et al (2015), *Variational Dropout and the Local Reparameterization Trick*, In Proceedings of the 28th Conference on Advances in Neural Information Processing Systems, 2015.
- [15] Olaf Ronneberger et al (2015), *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv.

## 10. Appendice

Considérons deux distributions de probabilités  $P$  et  $Q$ . La distribution  $P$  représente les données, les observations, ou une distribution de probabilités calculée avec précision. La distribution  $Q$  représente une théorie, un modèle, une description ou une approximation de  $P$ . La divergence de Kullback-Leibler s'interprète comme la différence moyenne du nombre de bits nécessaires au codage d'échantillons de  $P$  en utilisant un code optimisé pour  $Q$  plutôt que le code optimisé pour  $P$ .

$$D_{\text{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (5)$$

$$D_{\text{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx \quad (6)$$