

# État de l'avancement sur les data SWORD

## Padding vs Resize

J'ai voulu tester l'efficacité des deux méthodes sur 80 epoch car il y avait des témoignages qui affirmaient que les résultats sur 80 epoch étaient meilleurs que les résultats aux epochs plus faible (<10) pour le MRCNN.

Pour ces deux résultats, le training a été fait sur l'ensemble des couches (full) et pas seulement les dernières.

Sur l'image Figure 1, on voit en **rouge** le run Padding et en **gris** le run Resize. On constate que le rouge s'arrête au bout de 45 epoch car le programme a crash. Ce crash revient toujours autour de l'epoch 45 pour la méthode de Padding (et pas pour la méthode resize) avec une erreur liée à la mémoire GPU qui n'est pas suffisante (peut-être réglable en bidouillant les paramètres de TF ?) De plus, le Padding met un peu plus de temps de calcul que le Resize (second élément qui montre que le padding semble prendre plus de ressources ?)

On voit sur l'image qu'effectivement les valeurs des loss à l'epoch 10 sont moins bonnes que les loss à l'epoch 40 par exemple. Si on regarde **epoch\_val\_mrcnn\_class\_loss (9<sup>e</sup> plot)**, à 10 epoch on a 0.25 de loss pour la courbe grise (resize), et à l'epoch 80 on a 0.1.

Donc le réseau semble continuer d'apprendre et on pourrait donc laisser tourner les réseaux au-delà de 10 epoch pour voir les variations (c'est ce qui est fait pour tous les résultats à venir ici).

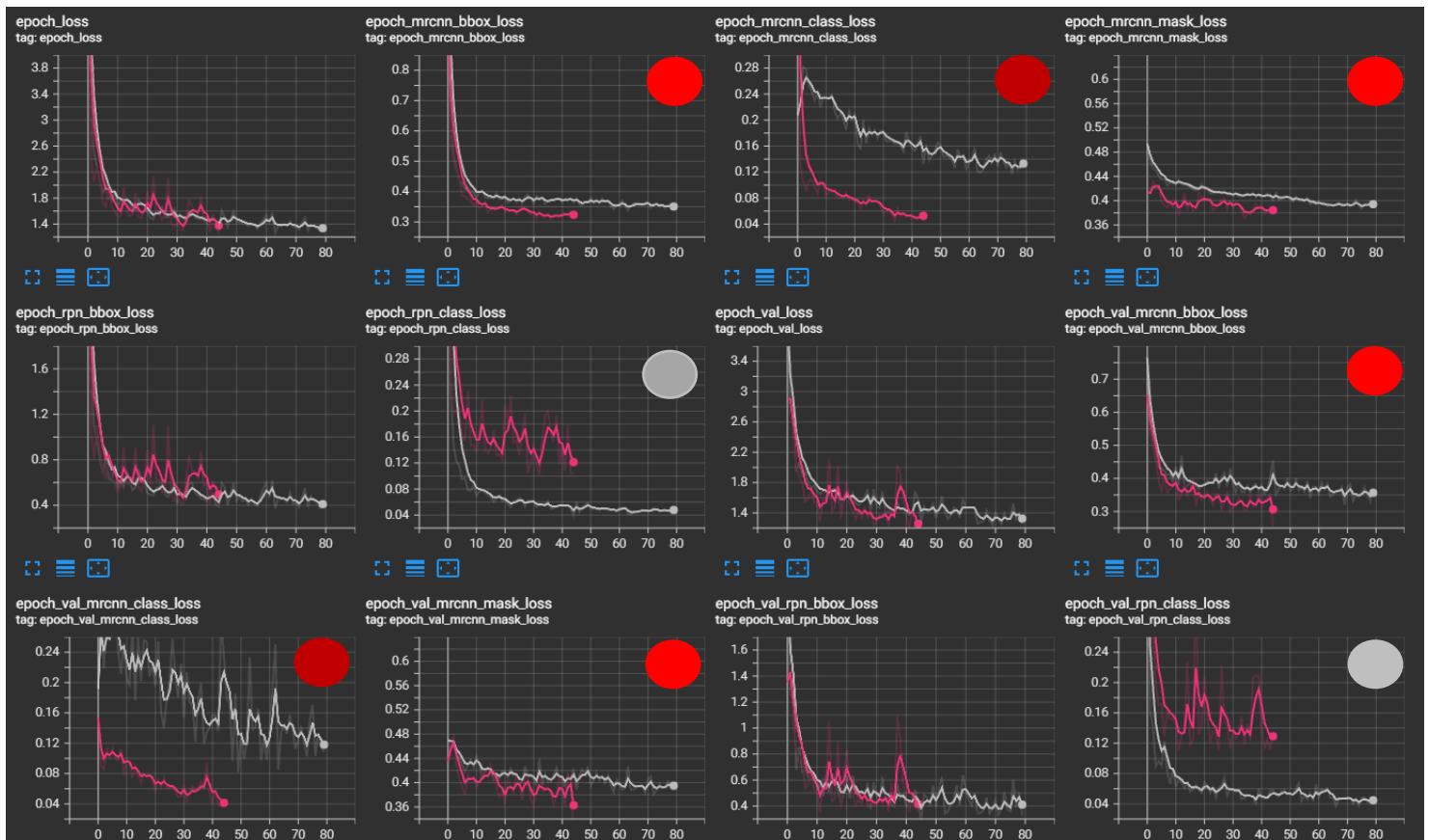


Figure 1 : Padding full vs Resize full

Maintenant, en comparant les loss entre les deux méthodes, on constate que le Padding est meilleur que le Resize sur les loss **epoch\_val\_mrcnn\_class\_loss**, **epoch\_val\_mrcnn\_mask\_loss** et **epoch\_val\_mrcnn\_bbox\_loss** (en train (6 premiers plot) et en val (les 6 suivants)) → Pastille Rouge La seule réelle différence concerne la loss **mrcnn\_class\_loss** (pastille rouge foncé), pour les autres loss (pastille rouge clair) ne sont que légèrement meilleurs pour le Padding.

La méthode Resize est meilleur que le Padding pour la pastille grise, soit la loss **rpn\_class\_loss** et c'est la seule loss où Resize est meilleur que Padding.

Afin de savoir quelle loss privilégier, il faut savoir qui sert à quoi :

**rpn\_class\_loss** : Capacité du réseau à **détecter s'il y a un objet (PU)** ou pas dans une région → Important

**rpn\_bbox\_loss** : Capacité du réseau à correctement localiser le RPN (pas trop compris ce qu'est le RPN). **Concerne aussi la précision de la boîte** → Important

**mrcnn\_bbox\_loss** : **Précision dans la position de la boîte** → Important

**mrcnn\_class\_loss** : Capacité du réseau à correctement classer l'objet détecté → pas important car si le réseau détecte un objet, ça sera forcément de classe PU (la seule)

**mrcnn\_mask\_loss** : Concerne l'accuracy du mask créé qui est dans la boîte → Je sais pas si c'est important, car si on a la bonne position de la boîte, on s'en fout un peu que le mask à l'intérieur soit bon ou pas non ?

Donc, la seule grosse différence en faveur du Padding et sur la loss **mrcnn\_class\_loss**, celle qui n'est pas importante. Mais sur les autres loss, le Padding est quand même meilleur (légèrement) que le Resize.

Le Resize est meilleur sur la loss **rpn\_class\_loss** qui fait partie des loss importantes. Cette méthode va donc mieux détecter la présence des PU sur la map mais sera moins bon sur les autres aspects comme qualité du mask produit ou dans la précision de la boîte.

On pourrait ainsi dire que la méthode du Padding est meilleure que la méthode du Resize malgré le fait que le Padding est censé moins bien détecter la présence des PU sur une map.

Voici quelques exemples de prédictions pour le réseau avec la méthode Resize :

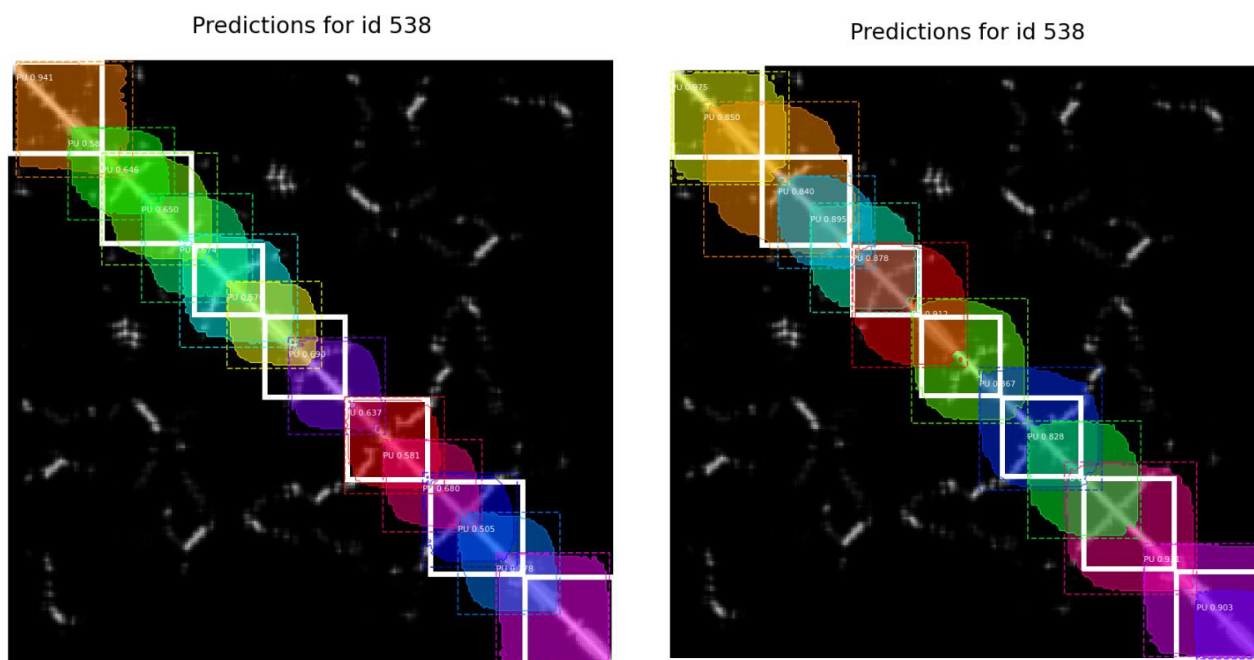


Figure 2 : Comparaison Resize vs Padding pour ID = 538

Le padding est censé moins bien détecter la présence des PU par rapport au Resize (gauche) mais on peut voir que ce n'est pas tout à fait le cas et que le Padding (droite) arrive quand même à les détecter et ce avec un bien meilleur taux de confiance pour chaque box (autour de 0.8 pour le Padding et 0.6 pour Resize). Les deux méthodes détectent 12 PU (alors qu'il y en a 7).

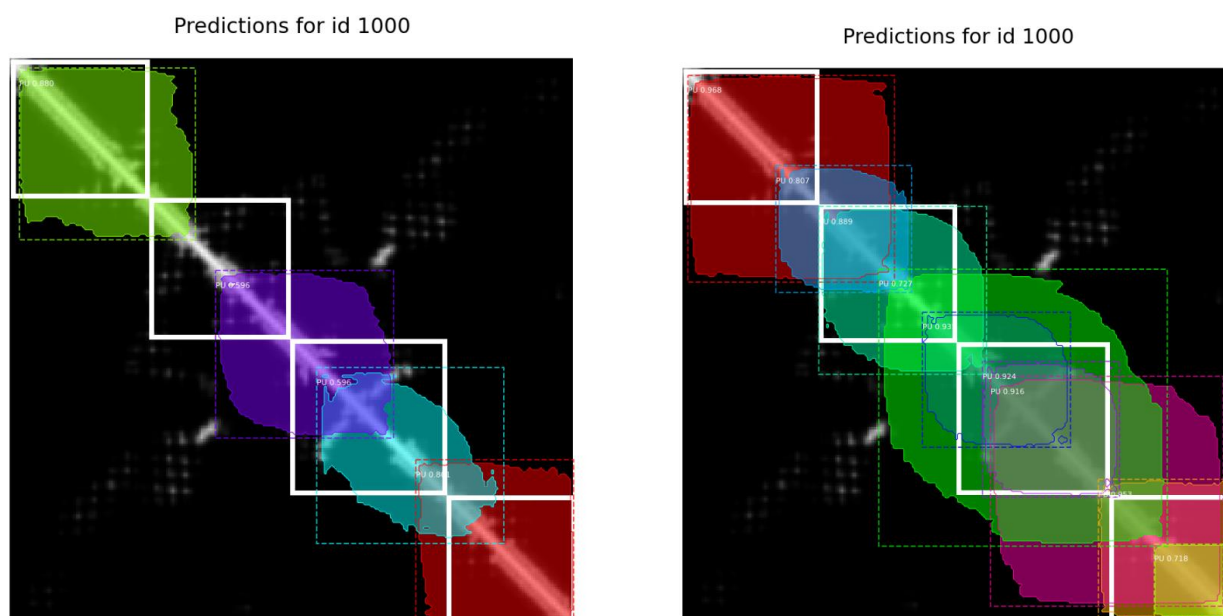


Figure 3 : Comparaison Resize vs Padding pour id = 1000

Sur ce 2<sup>e</sup> exemple figure 3, on voit la difficulté de la méthode avec Padding (droite) à repérer le bon nombre de PU (4) et en détecte 8. La méthode Resize (gauche) permet de détecter les 4 PU.

Un autre exemple avec la même observation :

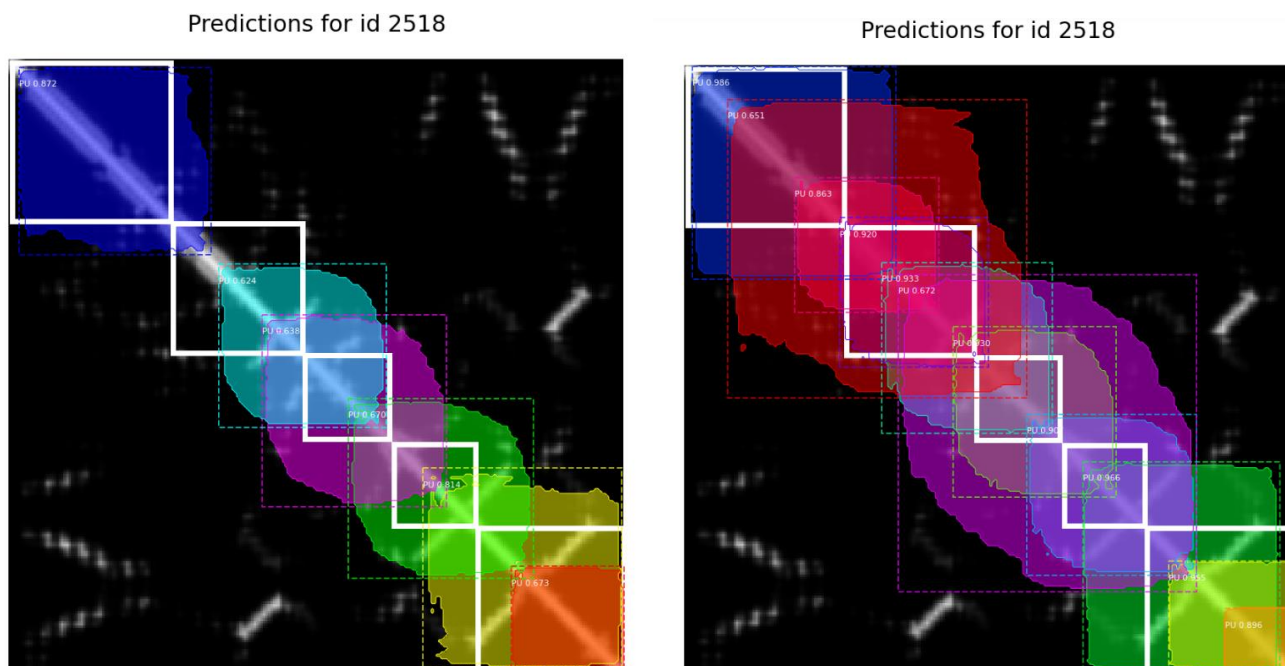


Figure 4 : Resize vs Padding

Ici aussi on voit que le Padding prédit mal le nombre de PU, il en donne 11 contre 6 pour Resize (La map a 5 PU). D'ailleurs, les box du Padding sont pas terrible ici.

La conclusion de ces observations est assez floue. Les plots (Figure 1) indiquent que le Padding aurait des résultats plus équilibrés mais les observations (Fig 2,3,4) montrent que le Resize semble être meilleur.

Je pense qu'une simple observation des box sur quelques protéines n'est peut-être pas suffisante. Il faudrait quantifier la qualité des box sur toutes les protéines. Il existe une mesure mAP (Mean Average Precision) donné dans le kit MRCNN mais elle ne fonctionne pas. J'essaierai de la recoder moi-même.

## Ajustement des poids

J'avais lu que pour améliorer certains aspects du réseau, il faudrait augmenter le poids des loss. Par exemple si on veut que le réseau détecte le bon nombre de PU, on devrait augmenter le poids de la loss qui concerne le nombre de PU sur l'image, soit la loss **rpn\_class**.

J'ai ajusté les poids des loss dites importantes (**rpn\_class**, **rpn\_bbox** et **mrcnn\_bbox**) en les mettant à 2 au lieu de 1 afin de voir si on obtient de meilleurs résultats.

Cela a empirer les résultats pour les deux méthodes. Peut-être qu'un poids de 2 est trop fort ?

## Full vs Heads

### Resize

Je teste ici la qualité des réseaux selon si on train sur toutes les couches (full) ou seulement les dernières (heads) avec des poids COCO loaded pour les autres couches pour la méthode Resize.

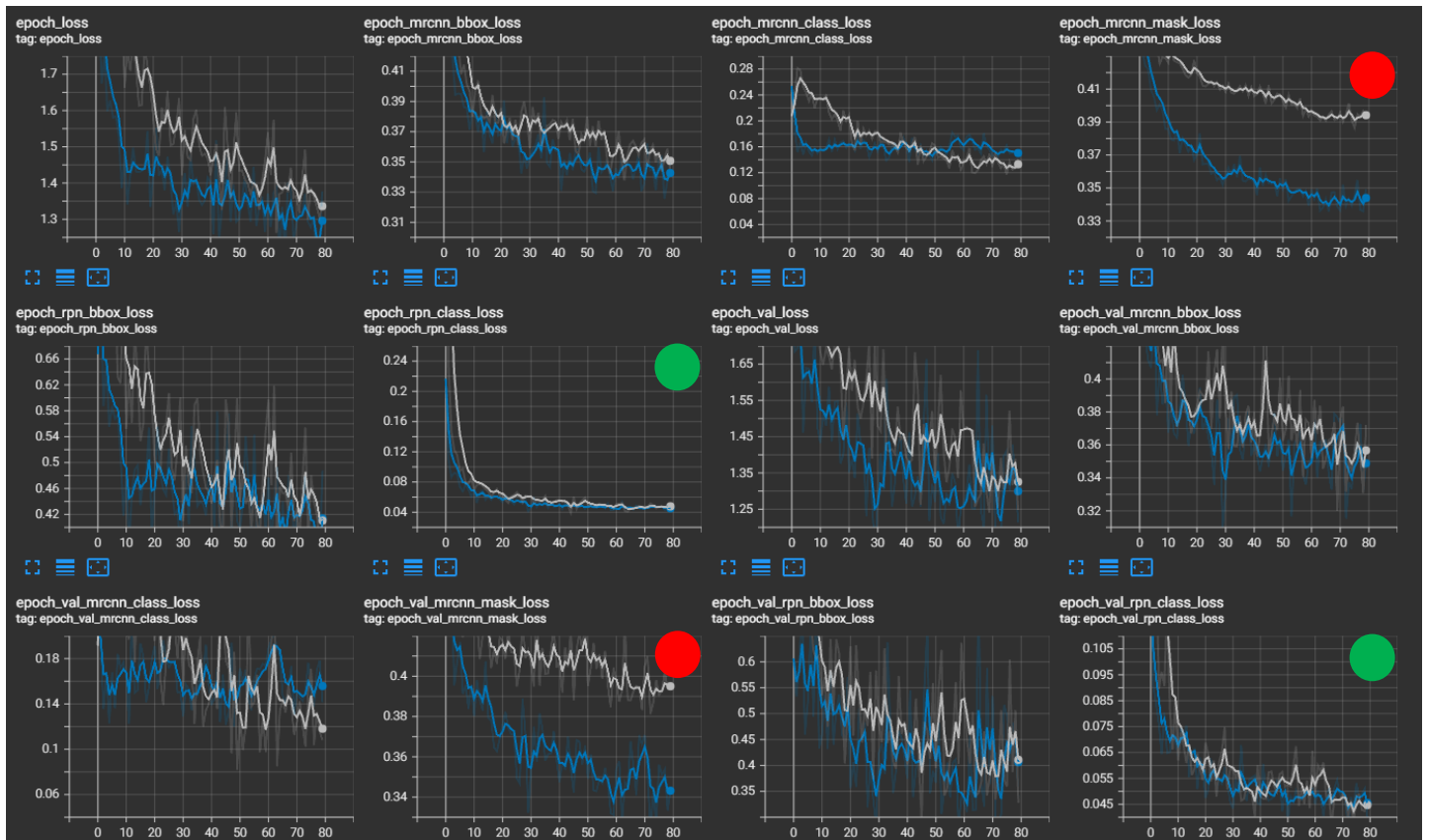


Figure 5: Plot Full (gris) vs Head (bleu) pour la méthode Resize

On observe sur ces plots que les valeurs sont très similaires entre Full et head pour toutes les loss sauf pour une loss : la loss **mrcnn\_mask** (pastille rouge) qui concerne la qualité des masks produit dans les boîtes (au niveau des pixels « bons »).

Pour le réseau entraîné seulement sur les dernières couches (bleu), il semblerait qu'il produise de meilleur mask que pour le réseau entraîné sur toutes les couches sans charger de poids au préalable (gris).



On peut voir cette différence sur des exemples :

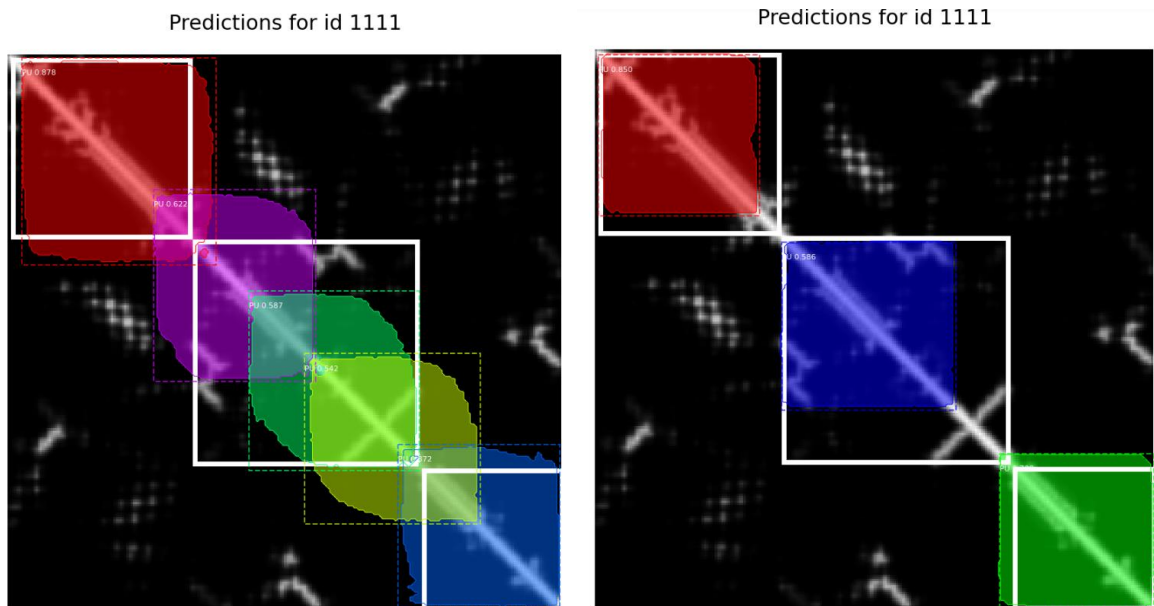


Figure 6 : Full vs Head (Méthode Resize)

On constate qu'effectivement, le réseau head (droite) donne de meilleur mask que le réseau de Full (gauche) qui eux sont moins rempli. Mais il donne aussi le nombre exact de PU (3) contre 5 prédit pour le réseau Full. Cette 2<sup>e</sup> amélioration n'était pas décrite dans les plots figure 5 qui est censée correspondre à la loss **rpn\_class** et qui est très similaire pour les deux réseaux (Fig 5 pastille verte).

Un autre exemple :

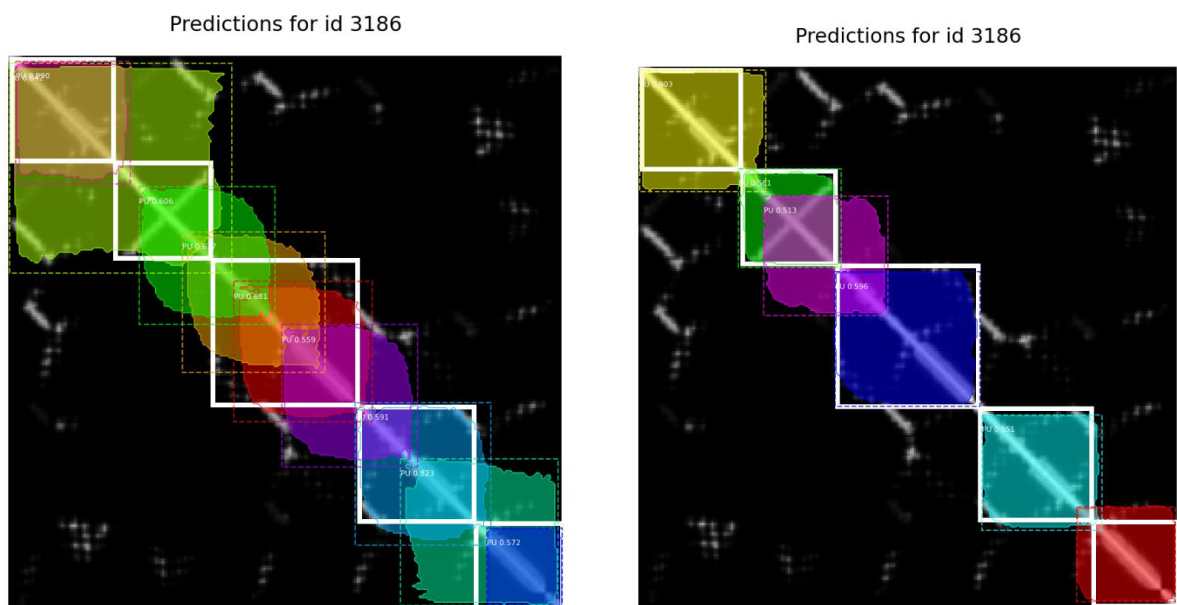


Figure 7 : Full vs Head (Méthode Resize)

Là aussi, les mask du réseau Head remplissent bien la boîte contrairement au réseau full, ce qui est attendu selon les plots (Fig 5). Et on observe également une différence dans la prédiction du nombre de PU qui est quasi correct pour Head qui en donne 6 contre 9 pour Full (Il y a 5 PU sur la map). On peut aussi constater la très bonne précision des box pour Head qui prédit quasi parfaitement les PU (sauf pour la box magenta qui est de trop).

On remarque aussi que la prédiction du réseau Full n'est pas mauvaise non plus si on regarde la position des bonnes boîtes (mais moins bonne que Head)

De ces observations, on pourrait ainsi conclure que le réseau entraîné seulement sur les dernières couches est meilleur que le réseau entraîné sur toutes les couches. On pourrait expliquer ceci par l'apport des poids COCO sur la qualité de la prédiction malgré le fait que COCO et PU ne sont pas les mêmes problématiques.

Ce sont ici des résultats **sélectionnés** afin de mettre en évidence les différences entre les deux méthodes sur des prédictions **plutôt bonnes**.

Pour le réseau Head, il ne prédit pas toujours aussi bien que Fig 7. Si on affiche un tableau 3x3 de prédictions sur des protéines aléatoires pour Head :

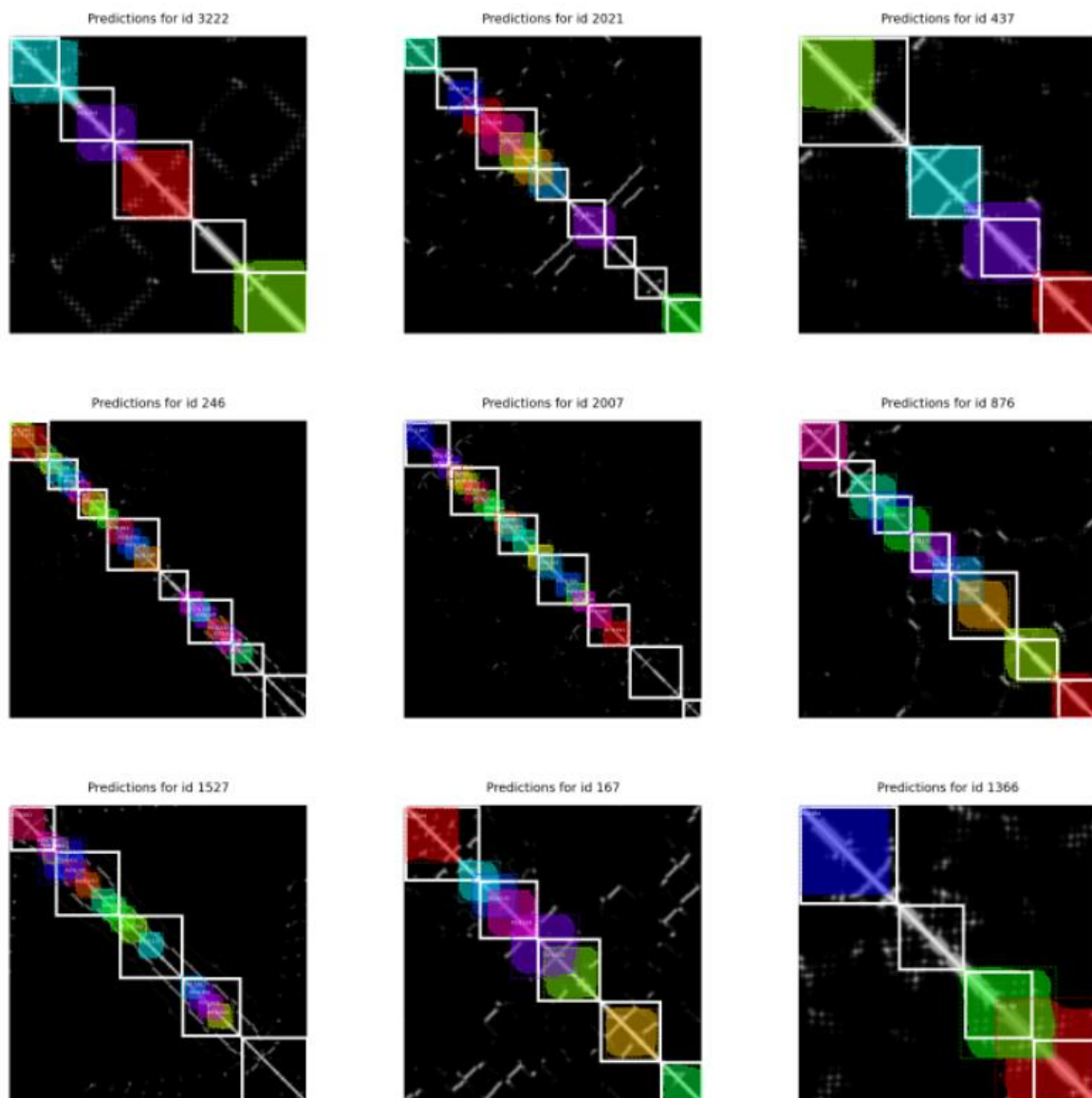


Figure 8 : Tableau de 9 prédictions pour le réseau Head

Il semblerait que ce réseau (et tous les autres réseaux que j'ai pu tester) a du mal avec les PU de petites tailles (par exemple id = 246 ou 2007) mais se débrouille pour les PU plus gros. Peut-être qu'en agrandissant l'image d'input, le réseau pourra obtenir plus d'informations ? (ou pas du tout)

On peut remarquer que pour l'ID 1527, les petits PU prédits sont peut-être dus aux « marques » sur la map dans ces zones et le réseau a probablement assigné ces marques à des petits PU.

## Padding

La même analyse est réalisée pour la méthode Padding. Les plots sont les mêmes que pour la méthode Resize (Fig 5), on constate presque les mêmes différences entre Full et Head, c'est-à-dire que le réseau Head a de meilleure perf pour mrcnn\_mask (mais moins prononcé en validation) et donc est censé produire de meilleur mask que le réseau Full.

On reprend les mêmes ID que précédemment afin d'également comparer Padding et Resize :

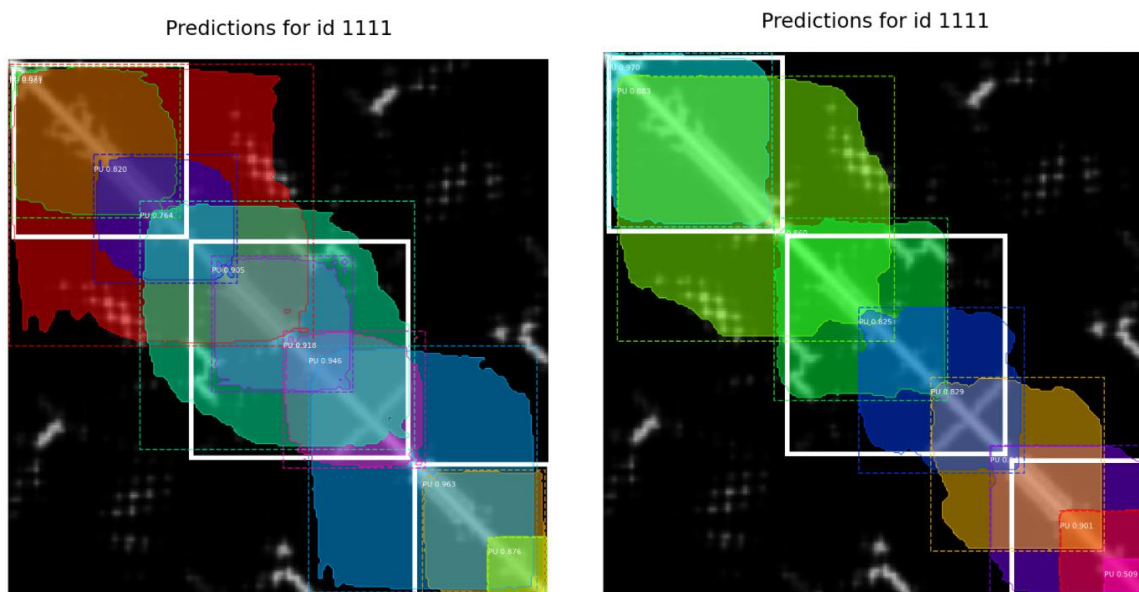


Figure 9 : Full vs Head pour la méthode Padding

La différence dans la qualité des mask produits n'est pas flagrante pour la méthode Padding entre Full et Head. Et globalement les mask produit ne sont pas terrible.

Pour les deux réseaux, la position des box n'est pas si mal mais la précision des box semble être moins bonne que pour la méthode Resize figure 6.



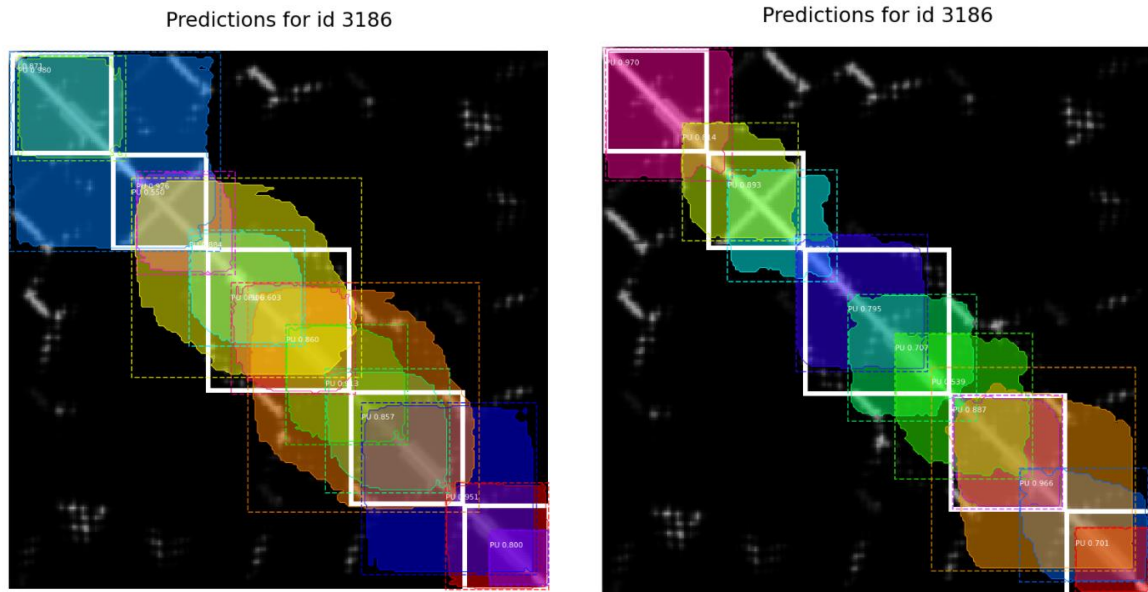


Figure 10 : Full vs Head pour la méthode Padding

Sur cet exemple, il est aussi difficile de voir une différence dans la qualité des mask entre Full et head. De plus, la position de certaines boites n'est pas trop éloignée des vrais PU.

Cependant, on retrouve ici la difficulté de la méthode Padding vu dans la 1ere partie qui est de détecter le bon nombre de PU. En effet, Full et Head affichent beaucoup plus que 5 PU. C'est également le cas pour le précédent exemple Figure 9.

Cette lacune n'est pas retrouvée avec la méthode Resize et les prédictions sont meilleurs (Figure 7) pour cette même map (id = 3186).

## Conclusion

Globalement, Resize semble être meilleur que Padding et Head est meilleur que Full. Le bon combo serait donc Resize Head en utilisant les poids de COCO.

De cette base, on pourrait essayer de trouver les paramètres optimaux de MRCNN pour améliorer les résultats et combler les lacunes de ce réseau à savoir les petits PU. (Agrandir l'image ?)

On pourrait aller jusqu'à 80 epoch mais il est vrai qu'au bout d'un moment, il n'y a plus trop de variations. On pourrait s'arrêter plus tôt.

Les plots seuls ne semblent pas être suffisant pour décrire les perfs d'un réseau car les observations peuvent donner des résultats qui ne sont pas trop en accord avec les plots des loss (étrange).

Il faudrait voir s'il y a d'autres poids disponible autre que COCO qui pourrait donner de meilleurs résultats (poids d'un réseau entraîné sur des formes carrés ?).

De plus, il faudrait une mesure beaucoup plus précise qui sera réalisé sur toutes les map et donner une accuracy globale. (à faire)

(Et tester l'architecture d'AlphaFold1)