Segmentation de la matière grise sur des IRM de cerveaux de nouveaux-nés.

Brève présentation du projet.

Notre projet consiste à segmenter la matière blanche sur des images IRM de cerveau de nouveau-nés, l’intérêt étant de pouvoir diagnostiquer tôt d’éventuelles maladie à partir de l’évolution du développement de cette zone. D’après la publication qui nous sert de support, la segmentation de cette zone est difficile à cause d’un contraste peu élevé. Celle-ci décrit un algorithme de segmentation des différentes zones en 5 étapes qui est spécifique aux cerveaux de nouveaux nés.

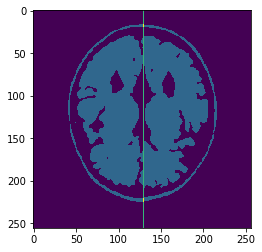
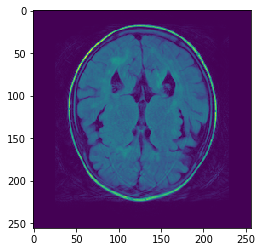
Les méthodes ont pour le moment été testées sur une image de taille 256x256.

Travail accompli.

# Partie 1 : Détection de la cavité crânienne et du cerveau + segmentation des hémisphères

Marianne a travaillé sur la première partie, qui consiste à obtenir le contour de la cavité crânienne, à extraire l’image du seul cerveau et à séparer les deux hémisphères. Pour obtenir le contour de la cavité, elle a essayé plusieurs méthodes (calcul du gradient de Sobel ou Laplacien). La plus concluante s’est révélée être un seuillage à partir de l’image filtrée avec un filtre médian (en gardant les 30% des points de plus haut niveau de gris), puis la suppression de tous les éléments à l’intérieur de la cavité.

Pour extraire l’image du cerveau, il a suffit d’implémenter une fonction renvoyant la plus grande composante connexe de l’image seuillée, qui correspond donc au cerveau.



*A droite, image originale ; à gauche contour ce la cavité crânienne, du cerveau et première séparation des hémisphères.*

Une première séparation des hémisphères a été réalisée en parvenant à implémenter une fonction qui trace une ligne entre les deux hémisphères.

# Partie 2 : Détection des noyaux gris.

Les noyaux gris ont un contraste qui est le même que le reste du cerveau, donc un seuillage par exemple ne s’avère pas être une méthode concluante ici. La publication propose une méthode qui consiste à réaliser des fermetures avec différentes échelles sur les images de cerveaux puis d’en effectuer la somme afin de mettre en valeur des minima locaux au niveaux des noyaux gris.

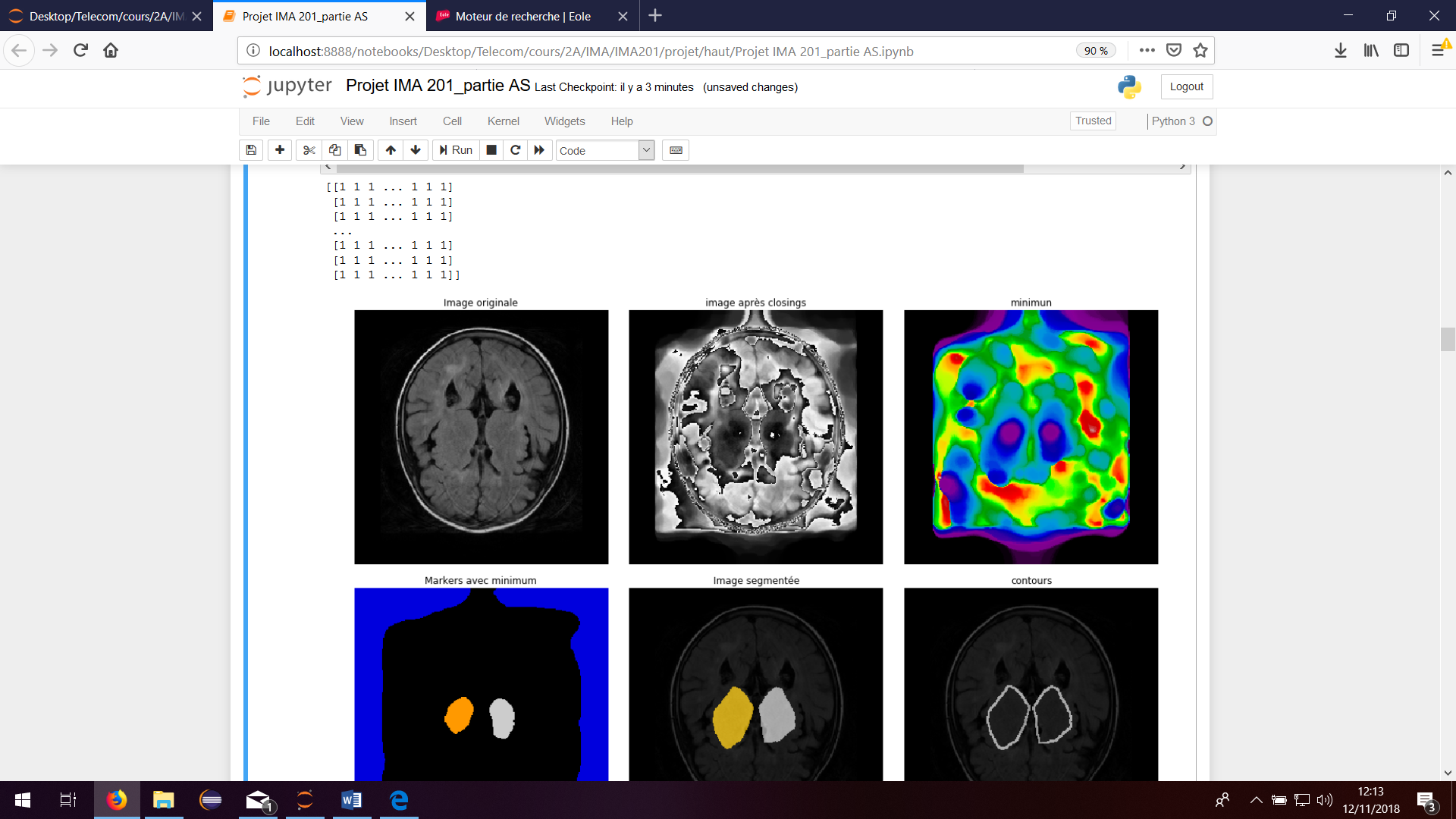
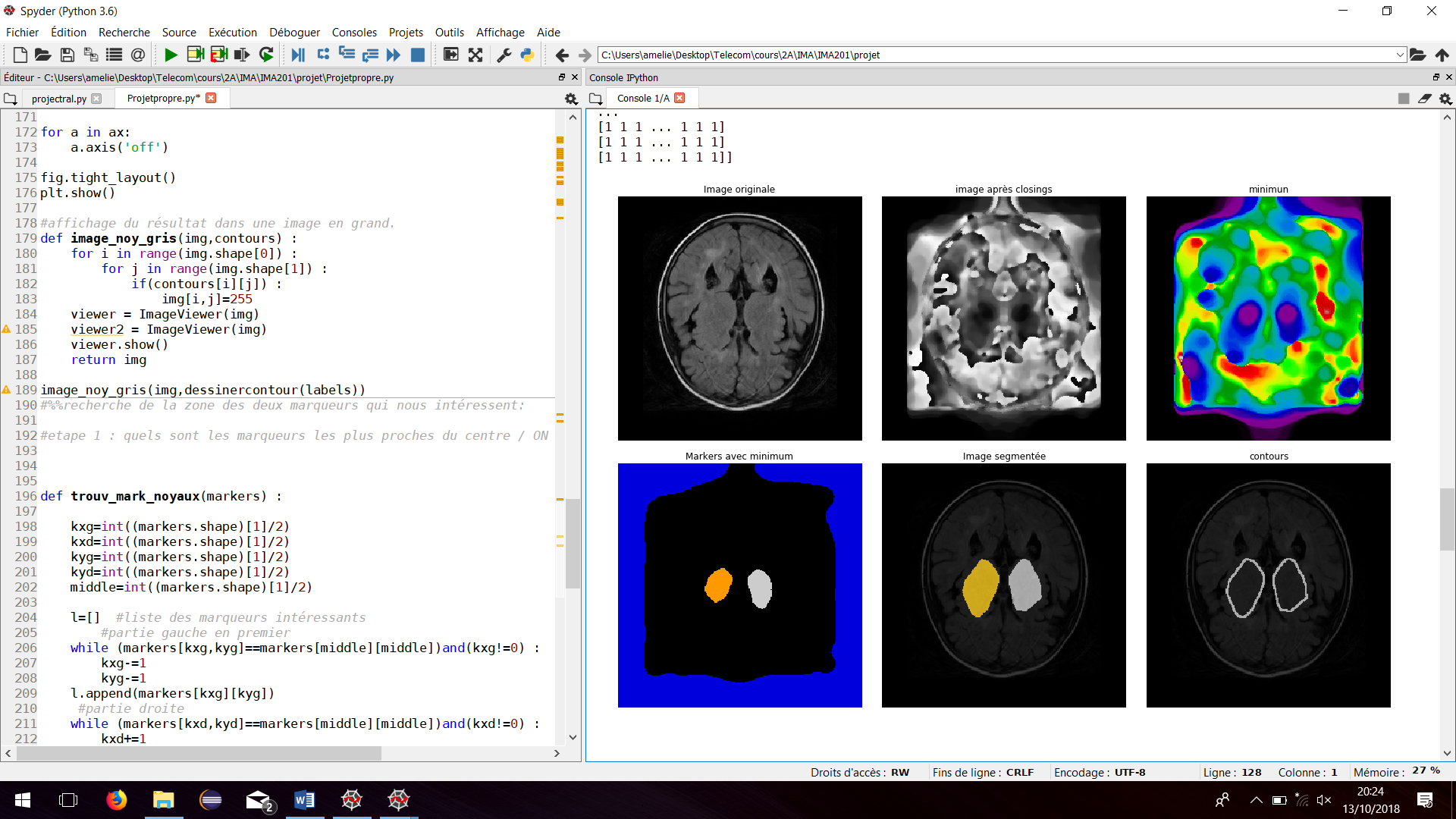
## 1.Somme des fermetures avec élément structurant d’échelle croissante.

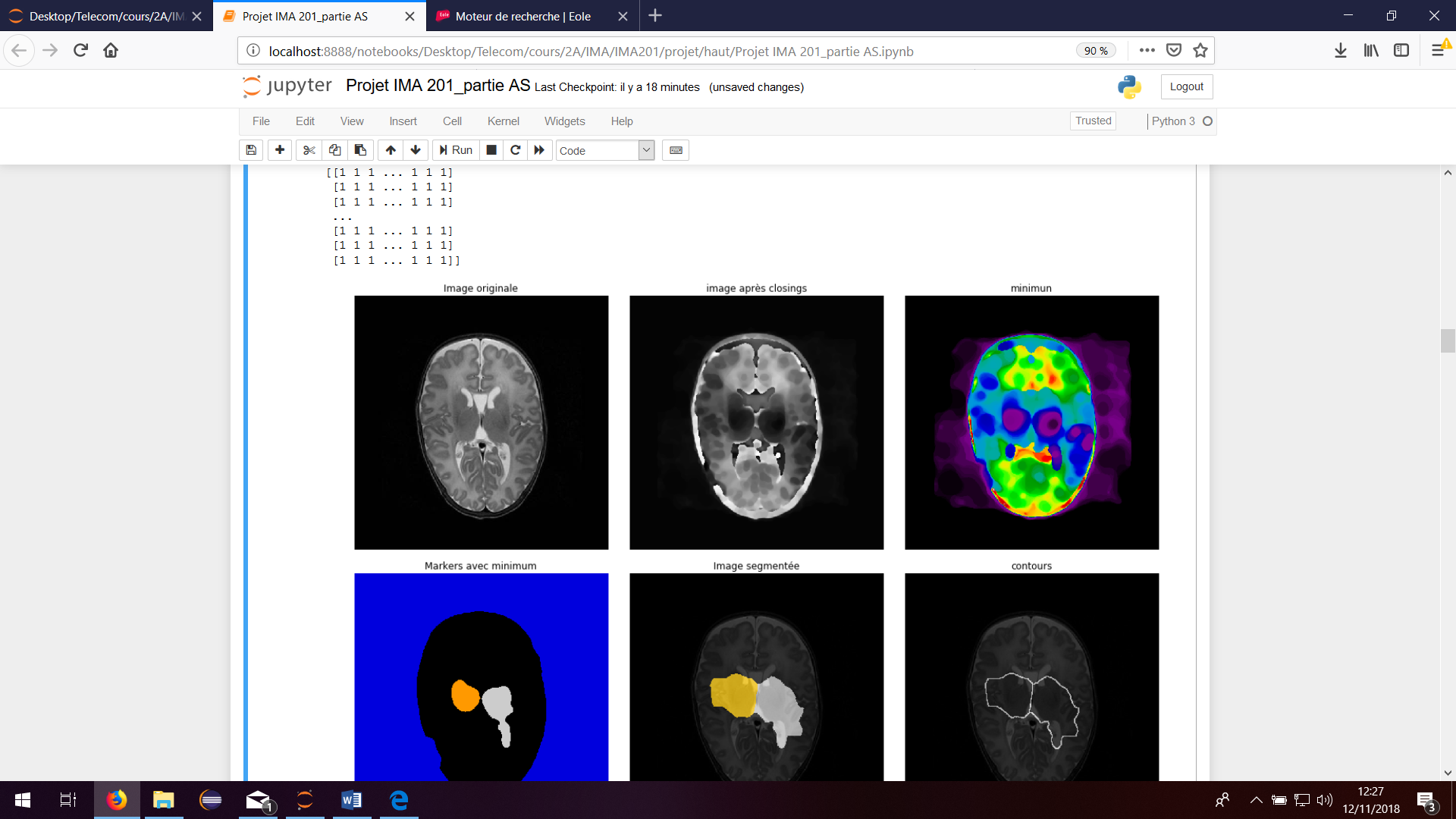
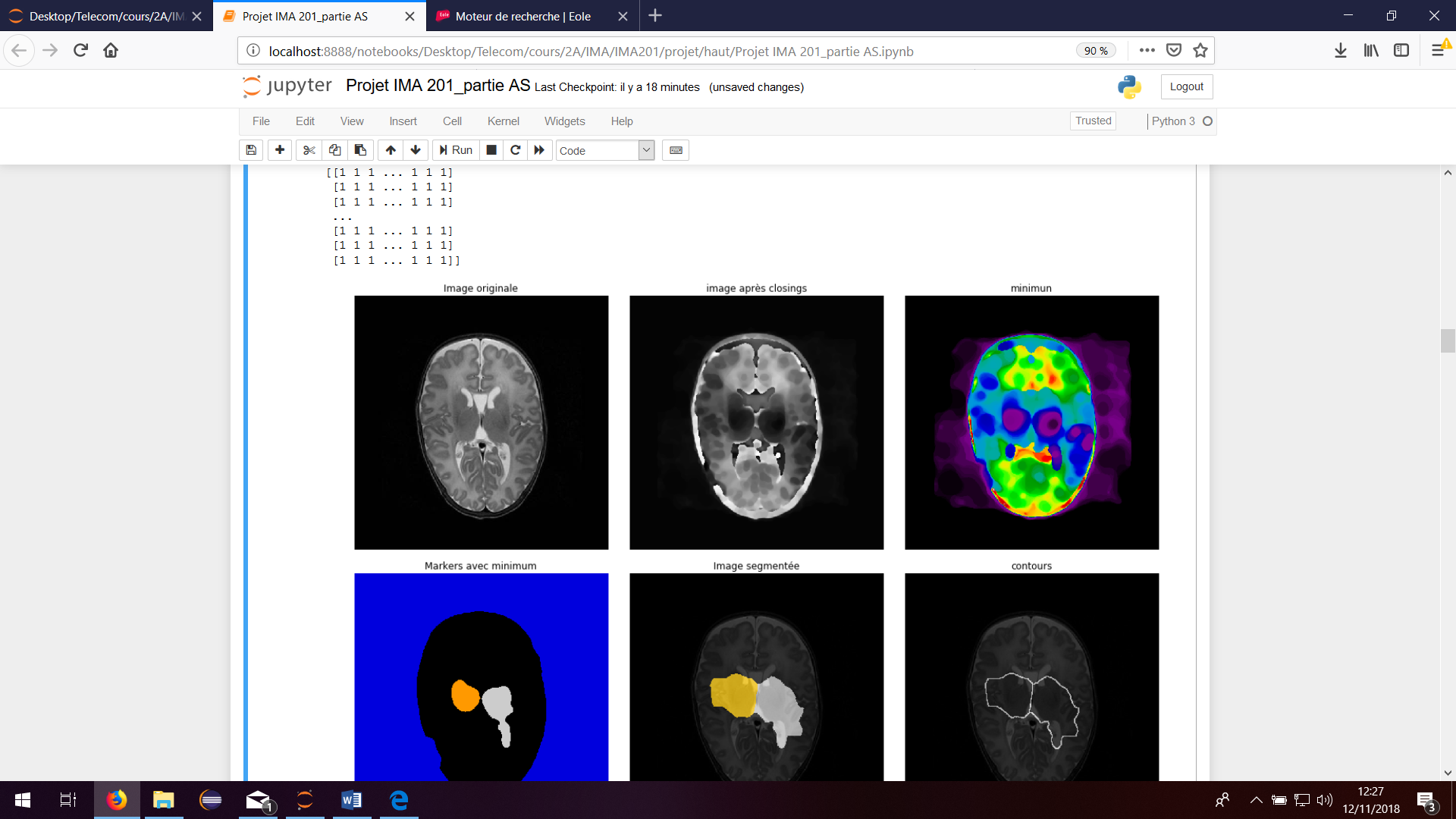
Amélie a donc réalisé la somme des fermetures d’images à l’aide d’ellipses de taille croissante. Cette opération est réalisée à partir de la fonction **incrclosing(k,n,pas,image).**Les paramètres *k, n* et *pas* correspondent aux différentes tailles d’élément structurant utilisées lors de l’opération : *k+1* correspond à la taille du plus petit élément structurant*, n* à la taille du plus grand élément structurant et *pas* à la différence d’échelles entre deux éléments structurants successif. Le choix judicieux de ces critères est primordial pour mettre en valeur des minimas au niveaux des noyaux gris (s’ils sont mal choisis, les minimas n’apparaissent pas). Ils permettent également d’éviter la saturation de certains pixels, due à la somme que l’on effectue sur toute les images, saturation qui conduirait à une perte d’information.

Pour la première image, les paramètres pertinents sont 50 pour le plus grand élément structurant, 1 pour le plus petit et un pas de 3 entre chaque taille d’élément structurant.

Cette somme induit un léger bruit qui pourrait gêner la détection du minimum, c’est pourquoi on applique un filtre médian de disque 3.

En appliquant un filtre médian pour débruiter et en testant plusieurs échelles différentes, elle est finalement parvenue à obtenir une image qui met bien en valeur les noyaux gris.



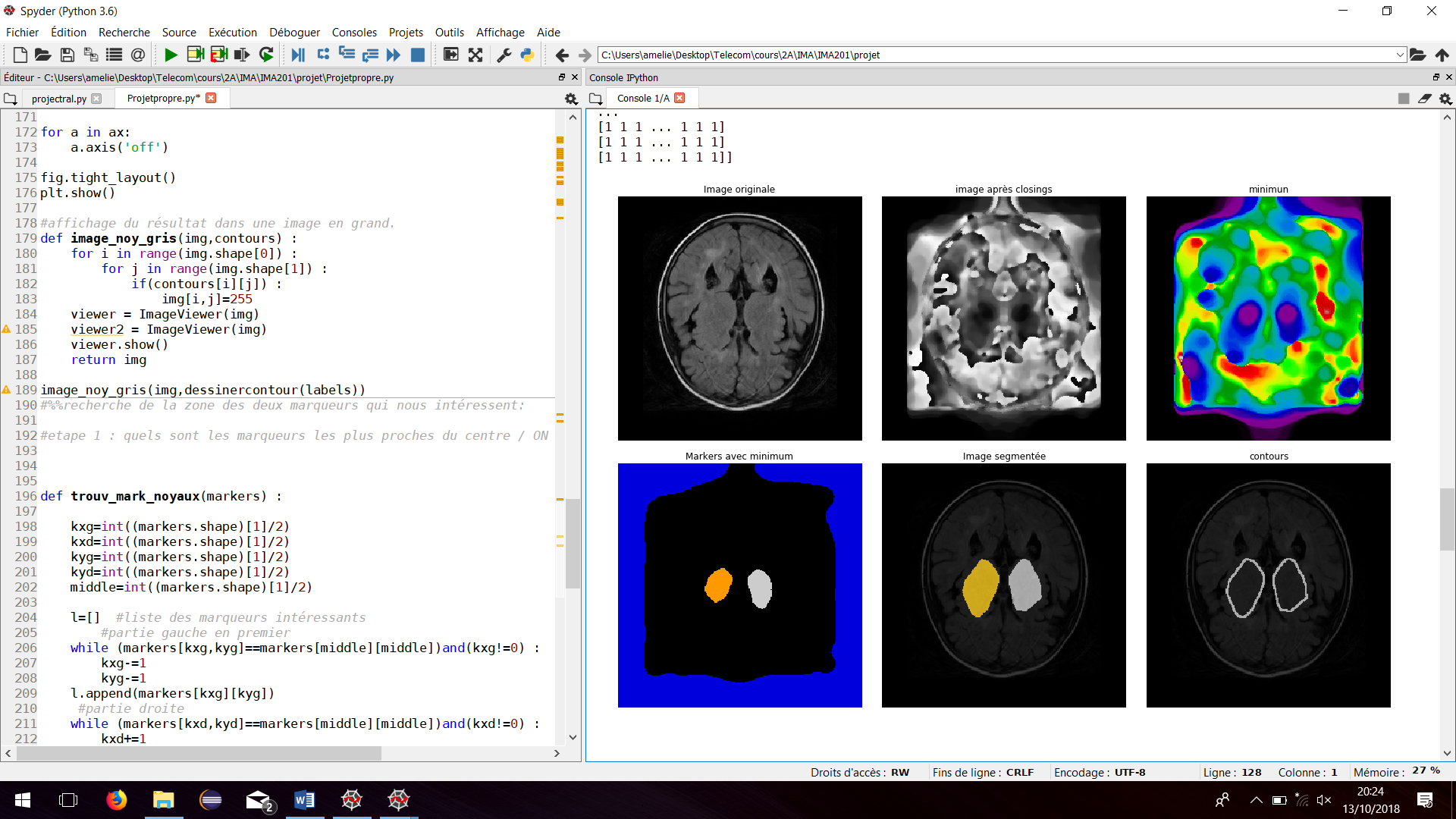
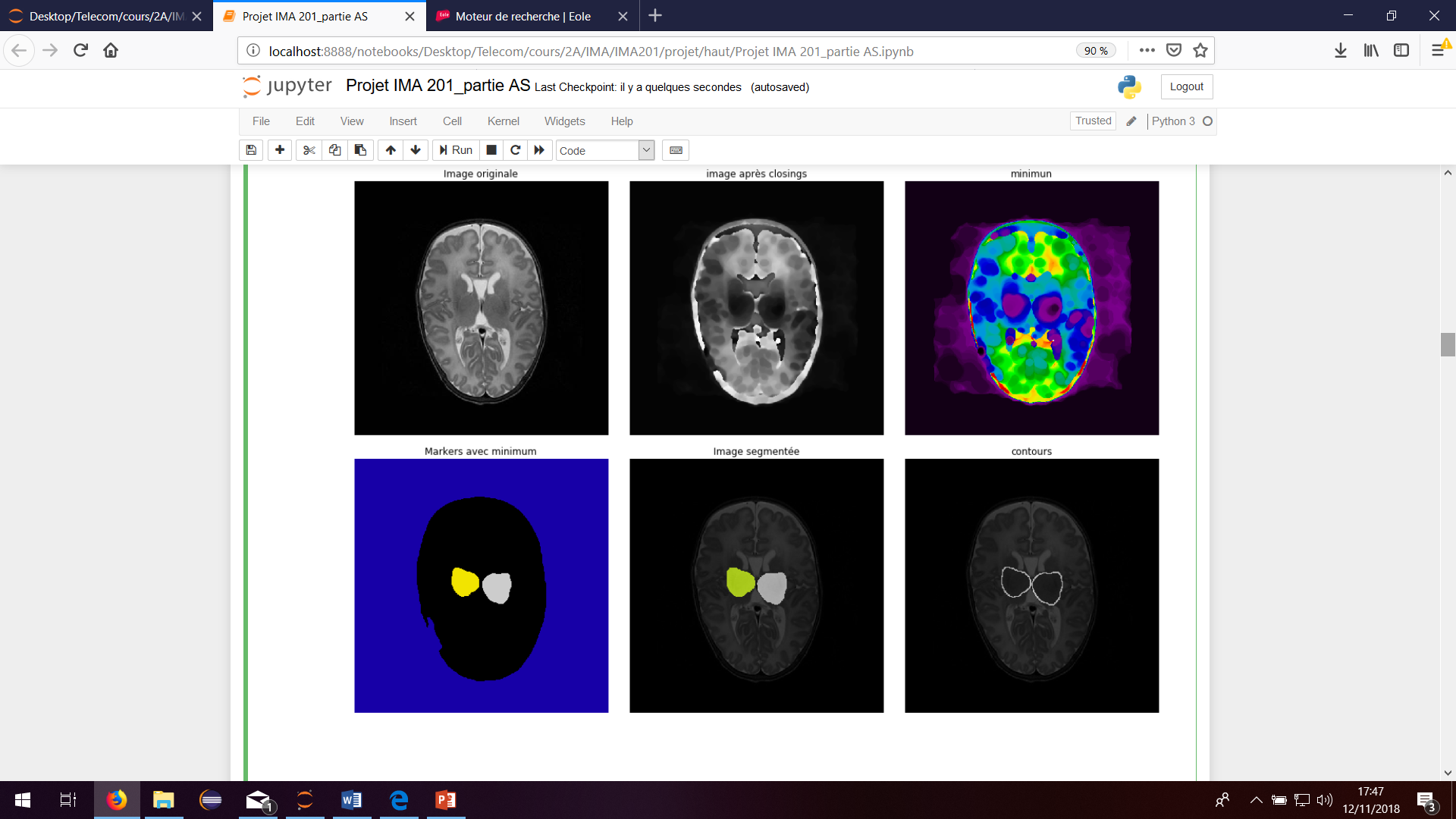
*Figure 1 : image originale Figure 2 : somme des fermetures d’échelle croissante. Figure 3 : Effet du filtre médian*

Avec les autres images, il est nécessaire de changer les paramètres afin d’obtenir un résultat satisfaisant. On applique la fonction avec k=10, n=60 et pas=20.

Le filtre médian est toujours de disque 3. On Obtient le résultat ci-contre.

## 2.Segmentation grâce à l’algorithme de ligne de partage des eaux.

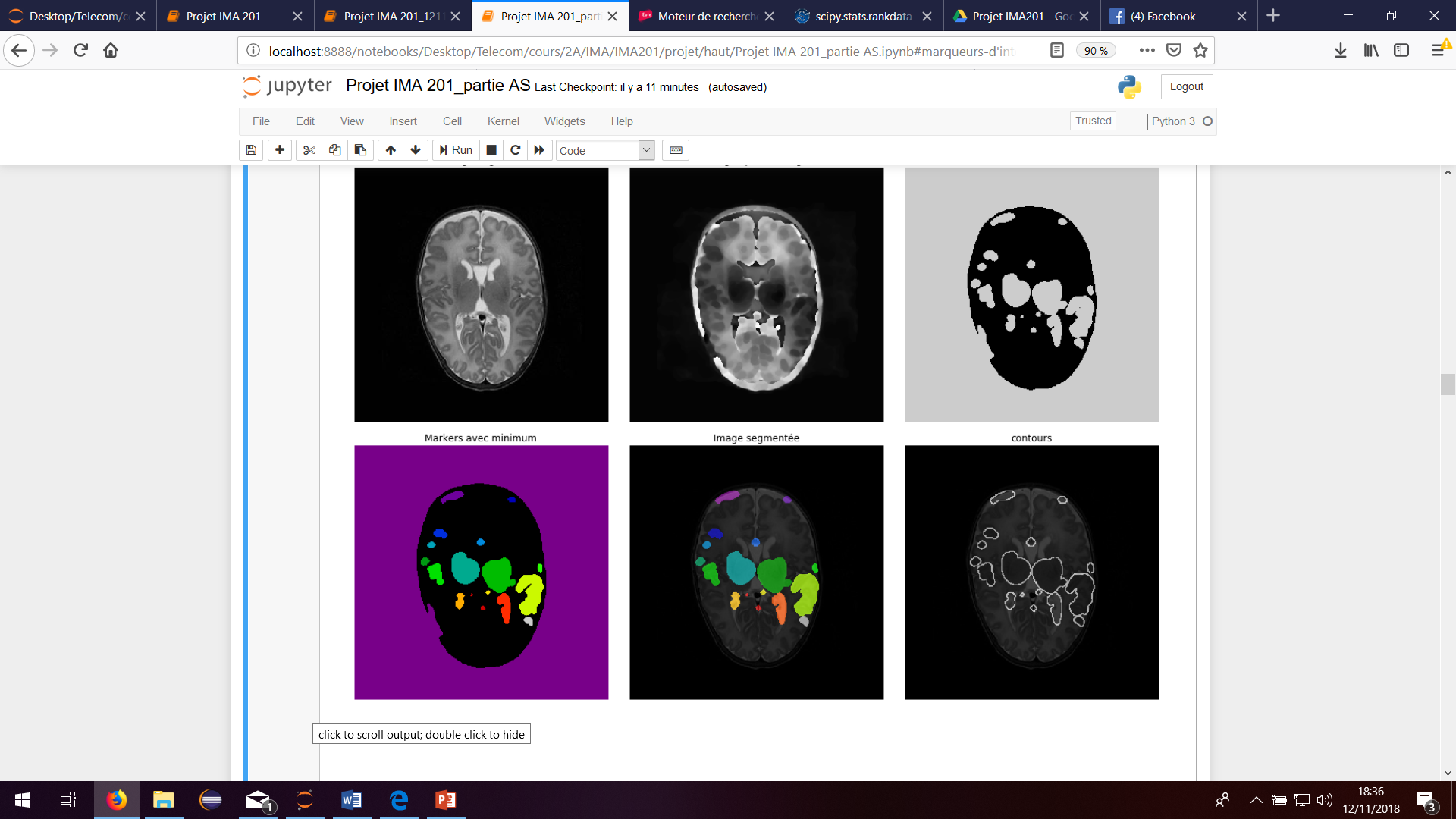
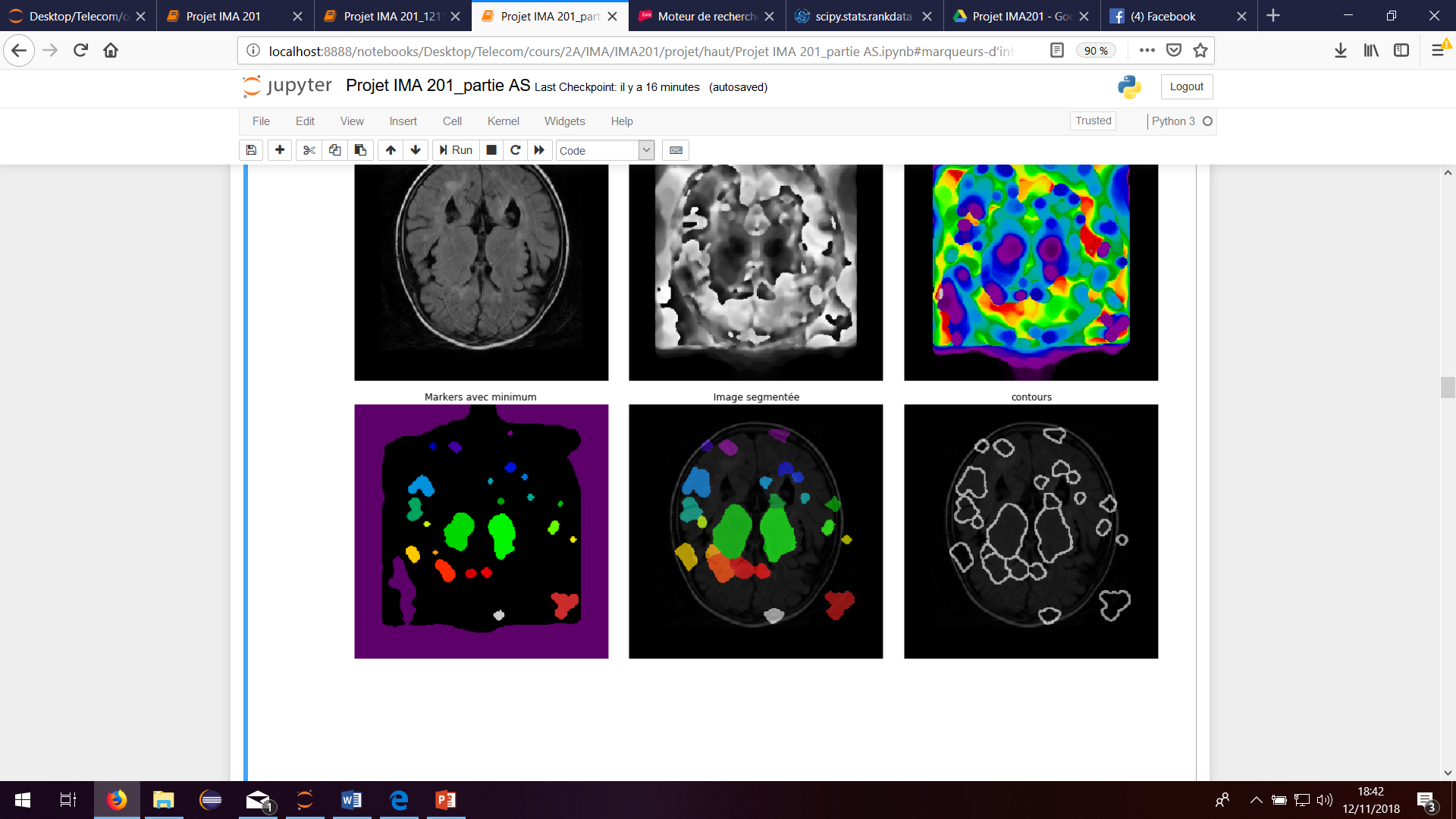
On implémente l’algorithme de partage des eaux sur l’image de la somme des fermetures. On utilise le minimum de cette somme pour trouver les marqueurs. Ici encore, on utilise différents paramètres qui varie selon les deux séries d’images.



*Figure 4 : Affichage du minimum pour les deux types d’image, avec comme critère rank.minimum<watershed\_seuil*

L’implémentation de l’algorithme est réalisé grâce à la fonction **segmentation(image,closing\_param,watershed\_seuil,bool\_afficher,seuil)** : *Closing\_param* correspond aux paramètres que l’on va utiliser dans la fonction incrclosing, *watershed\_seuil* correspond au seuil : mini= rank.minimum <watershed\_seuil, mini sera utilisé dans l’algorithme de partage des eaux comme l’image sur laquelle il s’effectuera.

#### 1)Détection des étiquettes des marqueurs d’intérêt et nettoyage des autres marqueurs.

Les marqueurs obtenus en imposant un seuil de 50 à rank.minimum ne sont pas uniquement les marqueurs des noyaux gris :

Fonction **trouv\_mark\_noyaux(markers,crane) :**

On se sert de la fonction **contour\_crane(image,seuil)** pour avoir le contour du crâne comme repère spatial. On utilisé également et la fonction **fond(image,seuil)** (qui ressemble beaucoup à la précédente)Grâce à la fonction **repere\_crane(crane)**, dans un premier temps, on obtient les 4 coordonnées extrêmes du crâne. On s’en sert pour éliminer les étiquettes et les marqueurs qui ne correspondent pas aux noyaux gris mais sont des minima locaux tout de même (et sont donc détectés par la fonction).

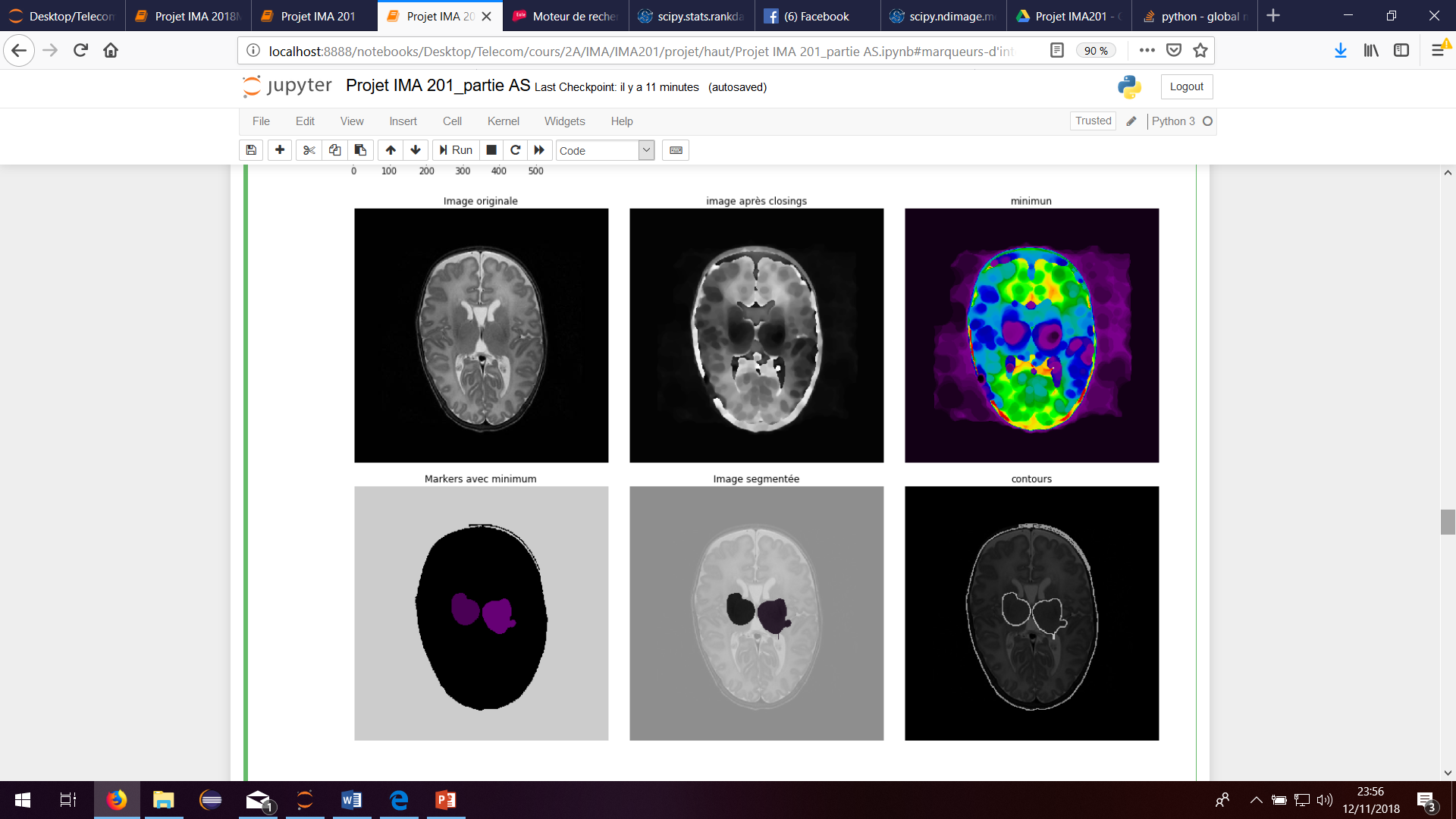
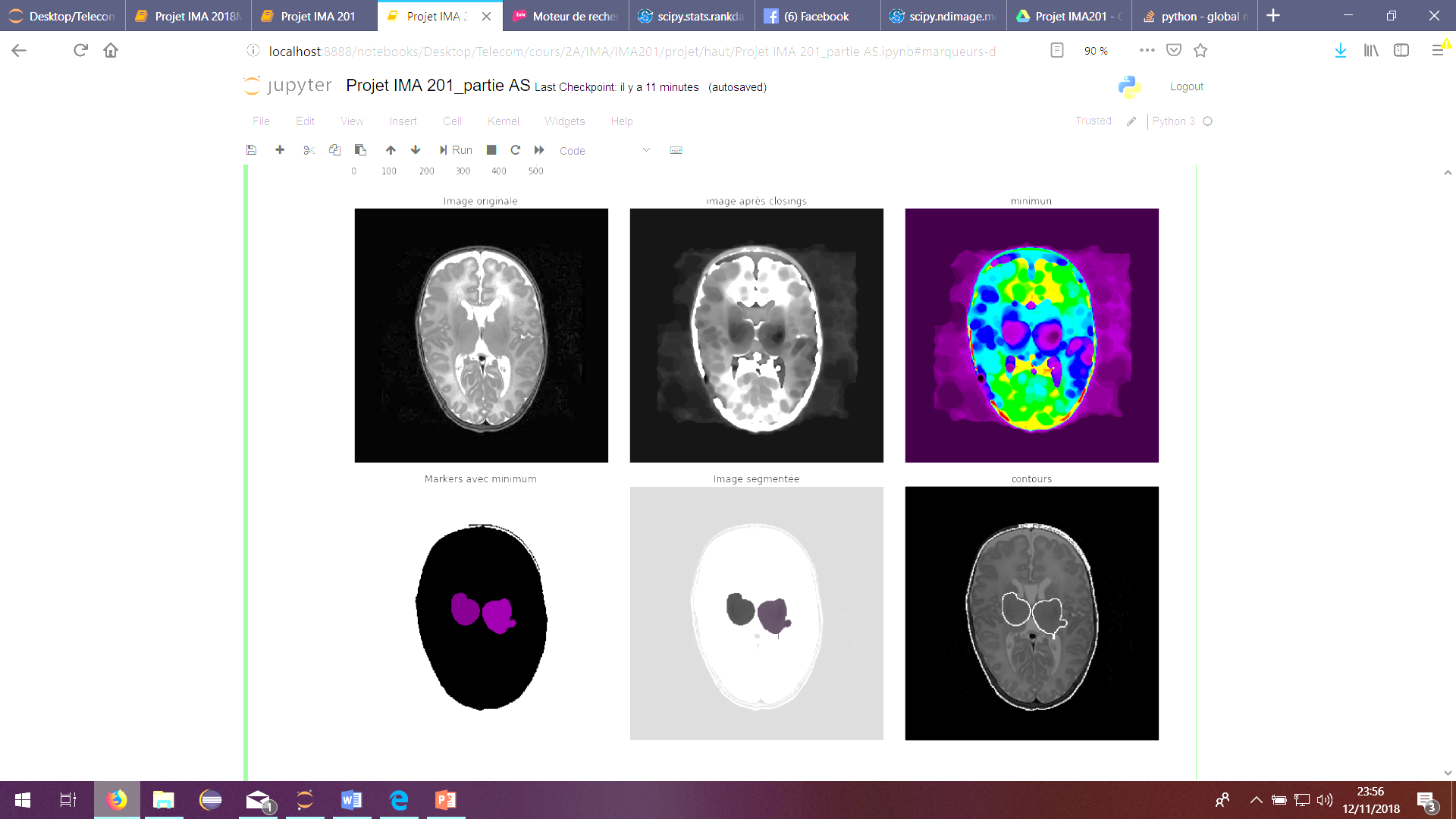
Les marqueurs des noyaux gris étant situés plutôt au milieu du crâne : on se place entre 2/5 et 3/5è de la hauteur, et entre 1/ 3 et ¾ de la largeur du crane, et on élimine ceux entre qui ne sont pas dans cette zone. En général, il reste encore plus de deux étiquettes, on élimine celles dont la zone contenant l’étiquette ne fait pas deux fois la taille du marqueur avec le plus de pixel. Si cela ne suffit toujours pas pour avoir au plus deux étiquettes, on élimine celle de plus grand indice (c’est en général la plus éloignée). Si au contraire, la liste ne contient qu’une seule étiquette, on cherche à ajouter l’étiquette du noyau symétrique à celui détecté.

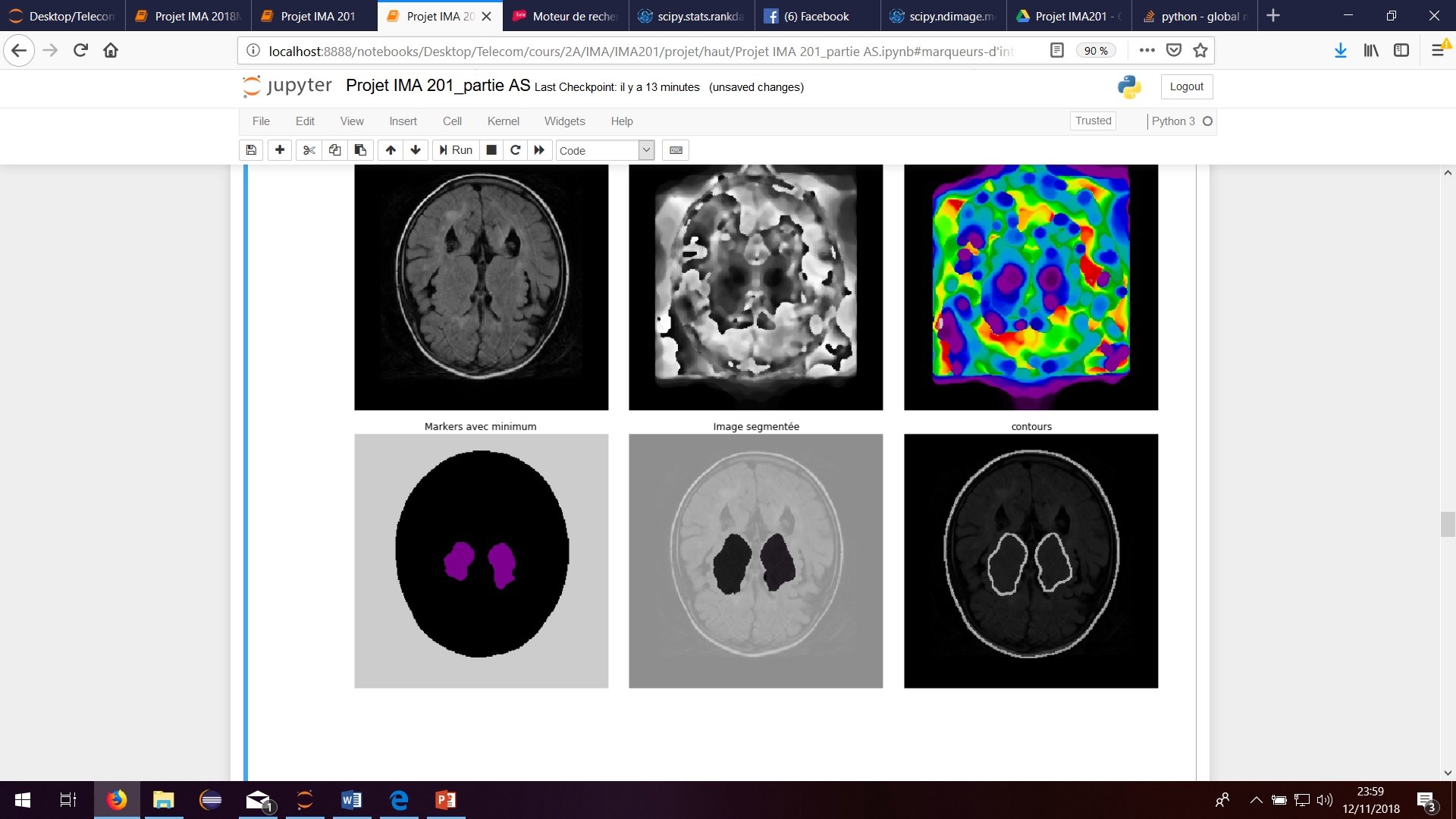
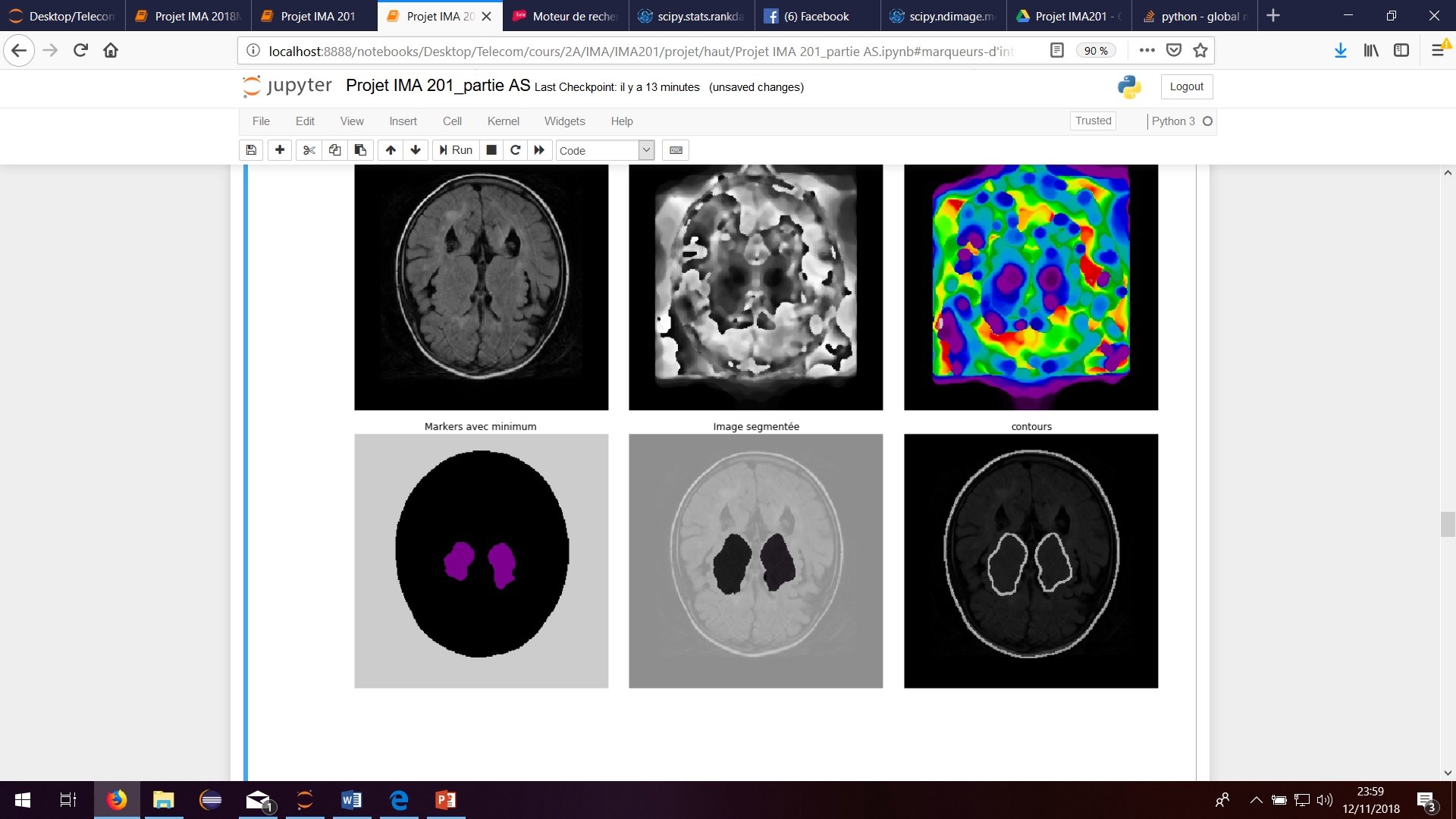
On élimine tous les marqueurs ne correspondant pas aux deux marqueurs d’intérêt grâce à la fonction **clean\_markers(markers,l),** *l* étant la liste des marqueurs d’intérêt.Grâce à la fonction **fond(image,seuil),** on obtient le fond de l’image, et l’on l’utilise comme marqueur : on met tout le fond à une étiquette donnée afin que cela donne une segmentation réussie.

#### 2)Segmentation.

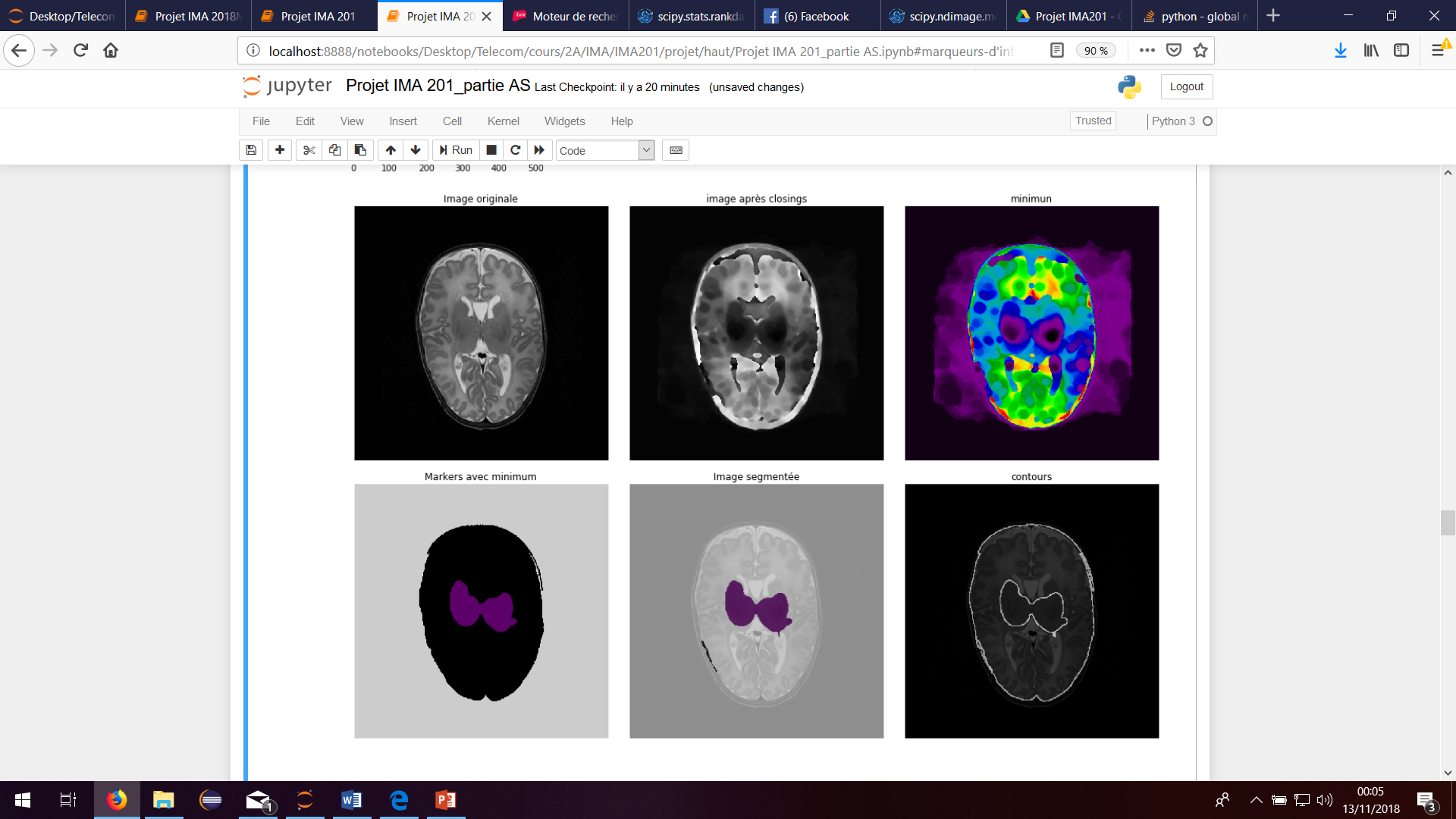
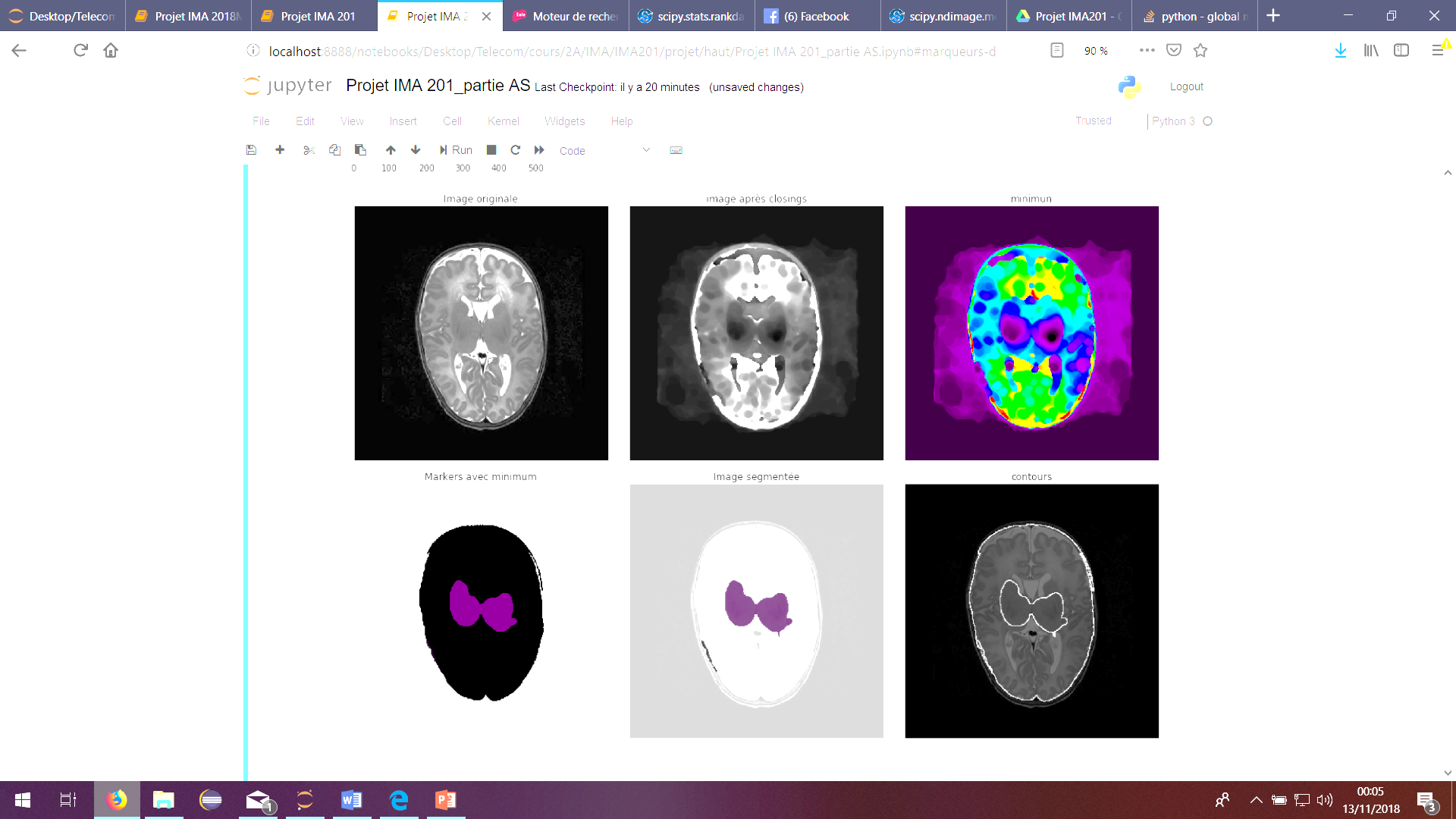
On applique ensuite l’algorithme de partage des eaux à partir du minimum. Enfin, grâce à la fonction **dessiner\_contour**, on affiche l’image avec les noyaux gris et le contour de la tête dessinés. On se sert également du contour du crâne pour l’afficher.

Le code fournit détecte les noyaux gris correctement pour les images T2014 à T2024 (sauf T2015 où les noyaux sont mal détectés). En revanche dans un nombre assez important de ces images, les noyaux gris sont reliés entre eux, mais, mis à part cela, a détection est bonne. Pour les images Image1Coupe10 et T2024 : ci-contre elle est excellente. On utilise pour cela les mêmes critères que décrit précédemment.

En revanche, pour les images T2025, T2026 et T2027, il faut changer les critères de fermetures.

Image T2024

*Image 1 Coupe 10*



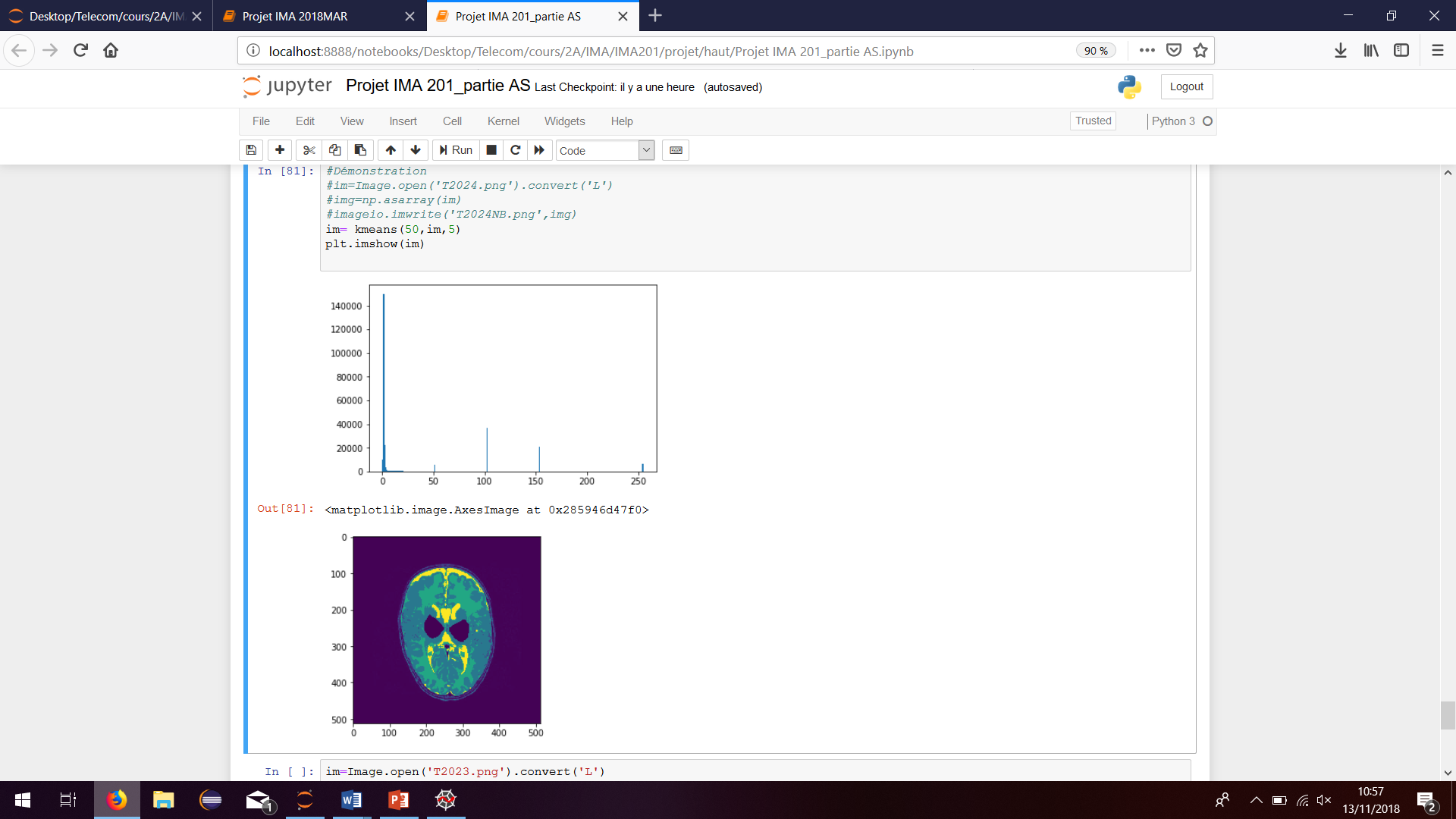
*Image T020 : noyaux reliés*

# Partie 3 : Segmentation de la matière blanche myélinée et non-myélinée par algorithme des k-Moyennes.

On implémente les k-moyennes avec deux critères pour la fin de l’algorithme : un critère de nombre d’itérations maximales et un critère de convergence.

On choisit comme critère de convergence, la moyenne des différences entre les positions des centres aux itérations k et k+1.

Plusieurs fonctions sont utilisées pour implémenter l’algorithme :

* **Initialize(k)** : initialise les centres de manière aléatoire dans k classes de niveaux de gris : on sépare simplement la totalité des niveaux de gris en k zones et les centres initiaux sont dans chaque classe.
* **affectation(centres,img)** : on affecte chaque point en fonction de son niveau de gris à la classe la plus proche.
* **nveau\_centres(centres,img,affectation)**: calcule les nouveaux centres comme le barycentre des anciennes classes.
* **kmeans(nbreit,img,k)** : effectue nbreit itération affectation puis calcul de nouveaux centres.

D’après la publication, la meilleure manière d’effectuer la séparation est de l’effectuer sur 3 classes ; avec le fond on applique l’algorithme sur 4 classes, après avoir retiré les noyaux gris grâce à l’algorithme de segmentation précédent et à la fonction **retirer\_noyaux\_gris(img,trouv\_noyaux\_gris, labels)**.