

## Rapport TP2 8INF804 : Vision artificielle et traitement d'images

### Introduction

Lors de ce travail nous essayerons de mettre en place un algorithme de segmentation qui permet d'identifier les grains individuels dans une image et extraire la couleur moyenne de chaque grain.

Notre démarche est la suivante, nous effectuons un prétraitement de l'image puis nous appliquons un algorithme de segmentation à l'image et enfin pour chaque segment identifié nous sauvegardons la couleur moyenne de ce segment dans un dataframe.

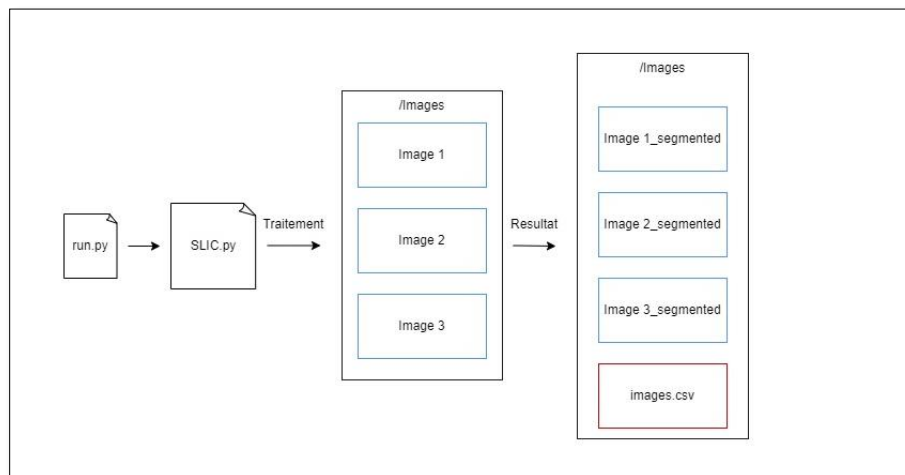
### Mode de fonctionnement

Notre code fonctionne à l'aide de **deux fichiers python**, le premier `run.py` est un fichier python qui parcourt l'ensemble des images dans le dossier et applique notre algorithme sur ces images et le second `SLIC.py` est le fichier python qui contient notre algorithme et le prétraitement effectué.

Pour exécuter le code `run.py` est utilisé uniquement, il prend en argument l'adresse du dossier qui contient les images. A la fin du traitement, les images segmentées sont sauvegardées et un fichier csv contenant le dataframe est créé.

Le code est exécuté **depuis le dossier** où se trouve le fichier `run.py`

### **Diagramme de fonctionnement**



### **Exemple d'exécution de code**

```
PS C:\Users\pau1e\OneDrive\Paul-Emmanuel\UQAC 1\8INF804 - Vision Artificielle\TP2_ATTE BERTRAND EVORA TOUGMA> python run.py "C:\Users\pau1e\OneDrive\OTHERS\Desktop\Images"
Processing file: C:\Users\pau1e\OneDrive\OTHERS\Desktop\Images\Echantillon1Mod2_301.png
Processing file: C:\Users\pau1e\OneDrive\OTHERS\Desktop\Images\Echantillon1Mod2_302.png
Processing file: C:\Users\pau1e\OneDrive\OTHERS\Desktop\Images\Echantillon1Mod2_303.png
Processing file: C:\Users\pau1e\OneDrive\OTHERS\Desktop\Images\Echantillon1Mod2_304.png
```

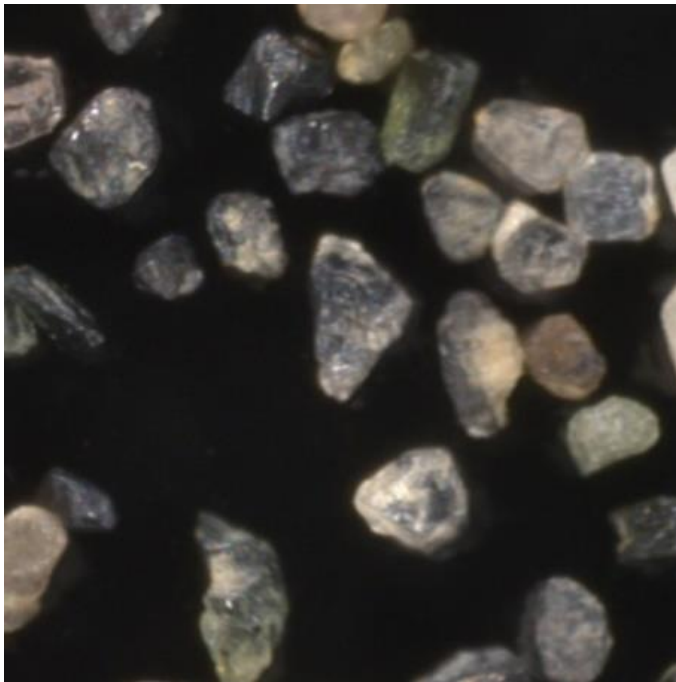
### Différentes méthodes de segmentation

Lors de ce TP différentes méthodes de segmentation ont été explorées dont contour based segmentation, watershed segmentation, SLIC segmentation et Canny Edge detector.

Chacune présentant des résultats intéressants, la technique finale de segmentation retenue est la segmentation SLIC car cette méthode permettait de mieux extraire les couleurs des différents grains.

Dans cette partie du rapport, pour comparer les résultats des différents algorithmes nous allons principalement utiliser la même image. **Lors du travail** la comparaison des résultats se faisait sur **l'ensemble des images** qui se trouvent dans le dossier.

L'image de comparaison des algorithmes :



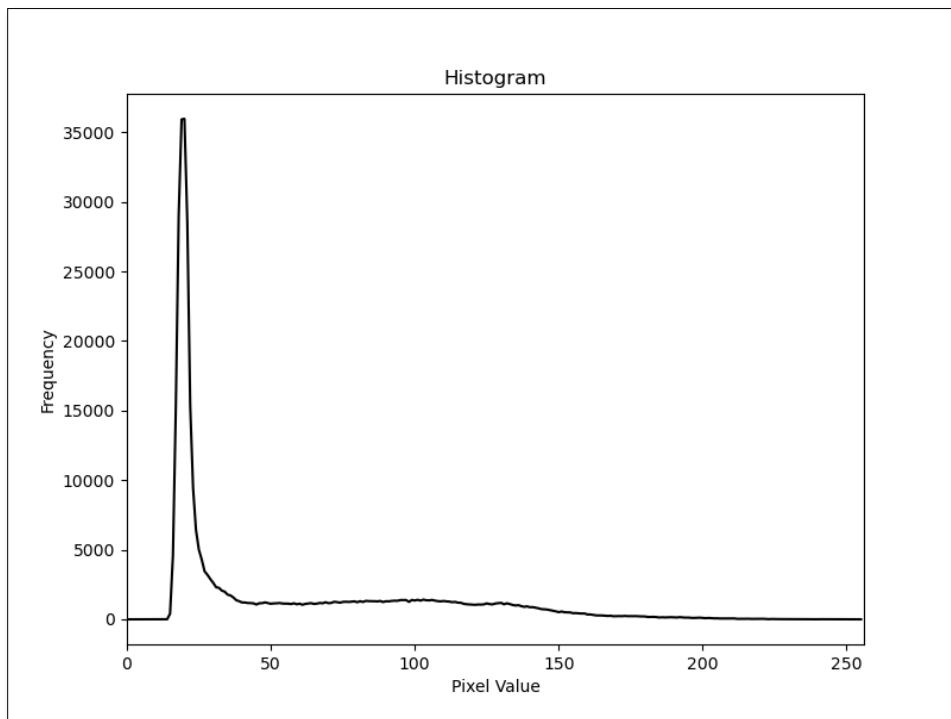
#### a. Contour based segmentation

Cette méthode de segmentation fonctionne en séparant les objets d'une image en différents segments en fonction la forme des objets trouvés.

##### Prétraitement de l'image

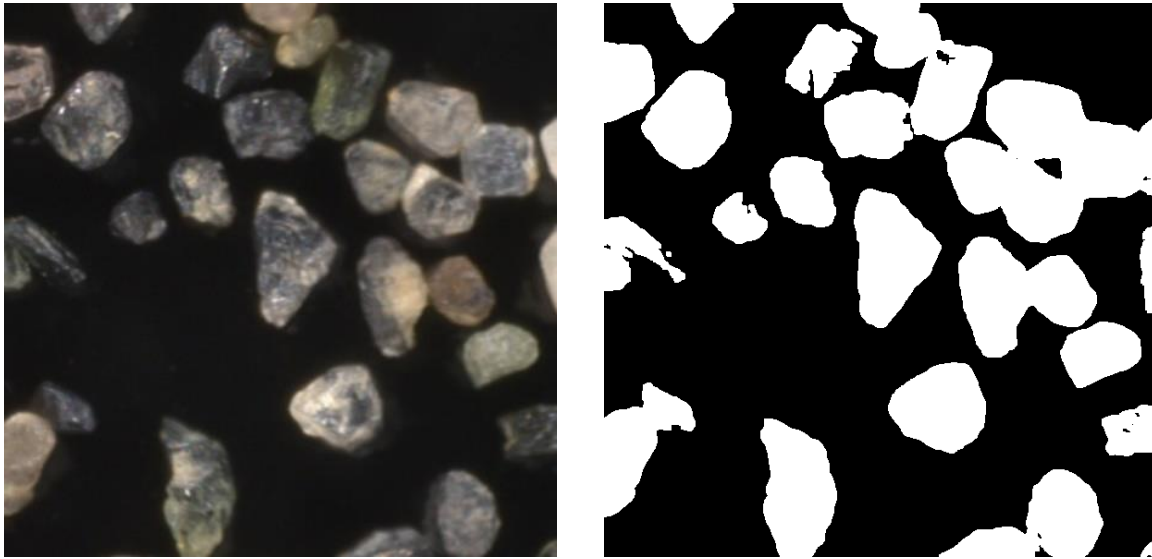
- Conversion de l'image en niveau de gris afin de limiter la taille des données traitées sans perdre de l'information essentielle
- Visualisation de la distribution de couleur afin de savoir à quelle valeur appliquer le thresholding
- Augmentation du contraste afin de rendre plus perceptible les contours des objets
- Appliquer un thresholding qui nous permet d'avoir un masque binaire séparant les grains de l'arrière-plan
- Appliquer un filtre d'opening pour enlever le bruit de l'image

##### Résultat preprocessing



Nous obtenons une distribution qui est largement centrée autour de 30, ainsi nous prenons comme valeur de thresholding 35. Nous appliquons d'abords l'augmentation de contraste et le filtre d'opening sur l'image.

### Résultat de thresholding

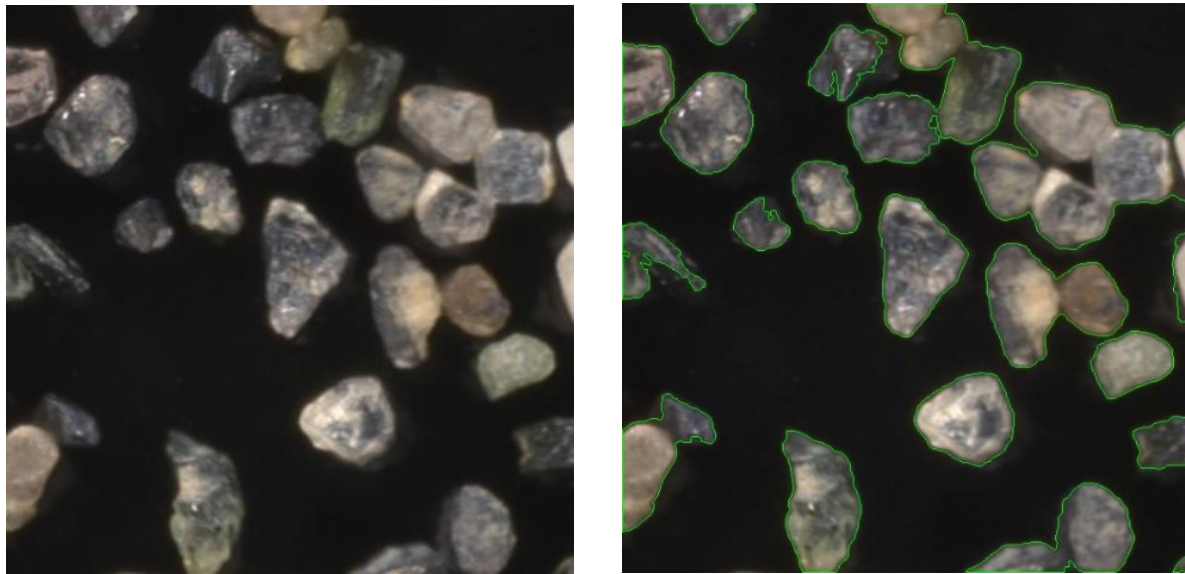


Le thresholding permet de dissocier les grains de l'arrière-plan.

### Application de l'algorithme

- On utilise la méthode findContours pour trouver les contours des grains dans l'image résultante du thresholding
- On parcourt l'ensemble des contours trouvés en évitant les contours de taille inférieure à un seuil
- On dessine le contour sur l'image originelle
- Pour chaque contour trouvé, on calcule la couleur moyenne dans l'espace de couleur BGR et on sauvegarde ces valeurs dans une liste

## Resultat de l'algorithme



La segmentation fonctionne bien pour les grains qui sont assez éloignés des autres grains, par contre lorsque les grains sont rapprochés ou se chevauchent l'algorithme les groupe tous ensemble.

La segmentation marche moins bien pour les grains qui ont une couleur qui tends vers le noir, dans ce cas le thresholding le mets dans l'arrière-plan.

## Resultat extraction de couleur

```
Test > Image_contour.csv
1  Moyenne de B,Moyenne de G,Moyenne de R
2  65.0,75.23970037453184,78.17602996254682
3  93.90455657848736,96.09346230043118,99.70236569164433
4  92.73317201935768,104.97877254729433,106.74076110866696
5  61.1502849002849,65.55591168091168,66.43304843304844
6  93.84861252360889,104.46505884062182,115.49731221850938
7  127.83067305134345,137.94573024740623,145.32535248736366
8  107.49181073703367,124.75432211101001,127.55914467697907
9  91.34557939914164,105.49553648068671,118.45605150214593
10 114.18657159833631,126.44325609031492,136.78312537136068
11 73.33097553787209,77.19658119658119,77.07515473032714
12 74.1654028436019,73.9872037914692,76.28578199052133
13 102.54650037630361,105.73970540802064,110.0685947747554
14 100.2865440464666,108.20062923523717,113.27105517909003
15 107.06574516406923,115.40347717127418,122.64186702169316
16 96.16874491732176,99.16210355109784,103.3423692057468
17 95.02382159148505,100.92929548910288,113.34870755195135
18 78.49401041666667,78.18307291666666,79.96276041666667
19 77.19372449859203,88.33090052295844,92.68312165967473
20 87.0166163141994,88.39425981873111,92.27240684793554
21
```

19 grains ont été identifiés et leurs couleurs ont été extraites, cependant vu le résultat de segmentation nous pouvons en déduire que des groupes de grains ont été considéré comme des grains individuels. Ainsi certaines lignes du dataframe correspondent à la moyenne des couleurs de **plusieurs** grains et non un grain individuel.

ATTE, BERTRAND, EVORA, TOUGMA

## b. SLIC Segmentation

Cette méthode de segmentation utilise du « clustering » une approche de machine learning non supervisé qui vise à grouper des données en fonction des similitudes trouvées par l'algorithme. Dans notre cas nous groupons les objets de l'image en fonction des similitudes de couleurs.

### Prétraitement

Afin de réaliser cette segmentation nous réalisons comme prétraitement les opérations suivantes :

- Conversion de l'image à l'espace de couleur LAB afin de faciliter la segmentation de l'image en fonction des différentes couleurs
- Augmentation du contraste de l'image afin de rendre plus perceptible les différences entre les grains

### Paramètres de l'algorithme

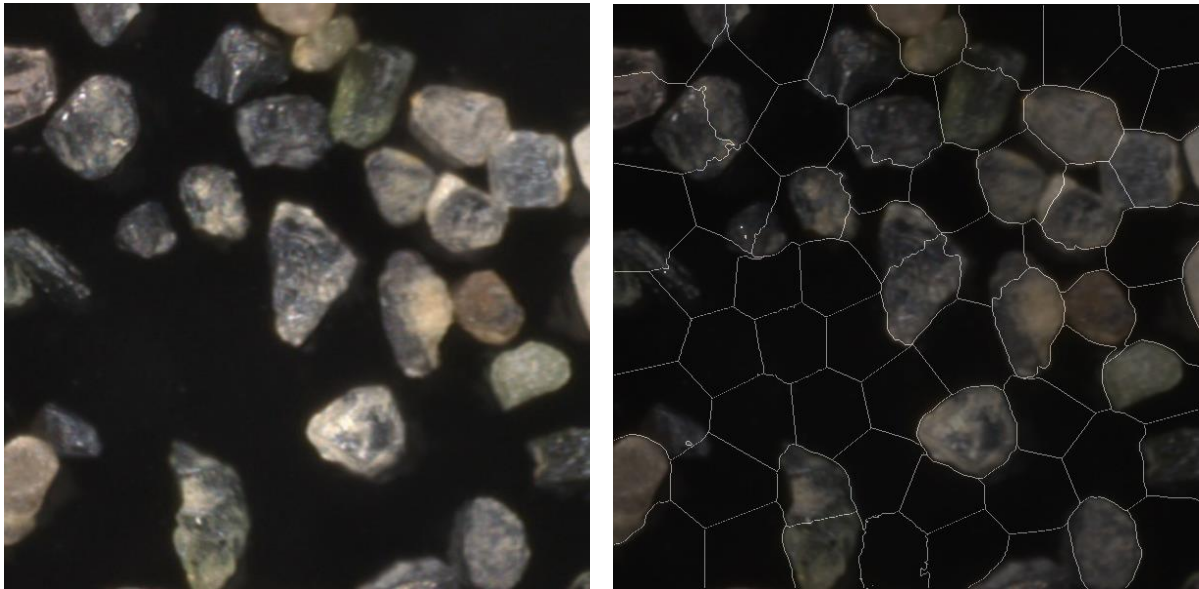
- Les paramètres essentiels de ce modèle : le nombre de segments et le « compactness »
  - o Le nombre de segments qui seront trouvés dans l'image a été ajusté en regardant l'ensemble des images dans le dossier.
  - o Le paramètre compactness définie à quel point les segments suivent des formes géométriques normales, lorsqu'il est élevé les segments trouvés ressemblent à des formes géométriques normales et dans le cas contraire les segments suivent plus les similitudes de couleurs trouvées
- Pour donner suite à différentes expérimentations les valeurs de paramètres qui donnent des résultats satisfaisants sont un compactness de 30 et nombre de segments de 75.

### Fonctionnement de l'algorithme

- On applique l'algorithme SLIC pour trouver l'ensemble des segments dans l'image
- On parcourt ces différents segments **en omettant** les segments qui ont une couleur moyenne trop sombre car nous ne voulons pas les segments de l'arrière-plan qui ne contiennent aucun grain
- Pour chaque segment on dessine le segment sur l'image originelle et on ajoute la couleur moyenne à un dataframe
- On remplit le segment sur l'image originelle avec la couleur moyenne trouvée afin d'avoir une confirmation visuelle que le calcul de moyenne se fait correctement

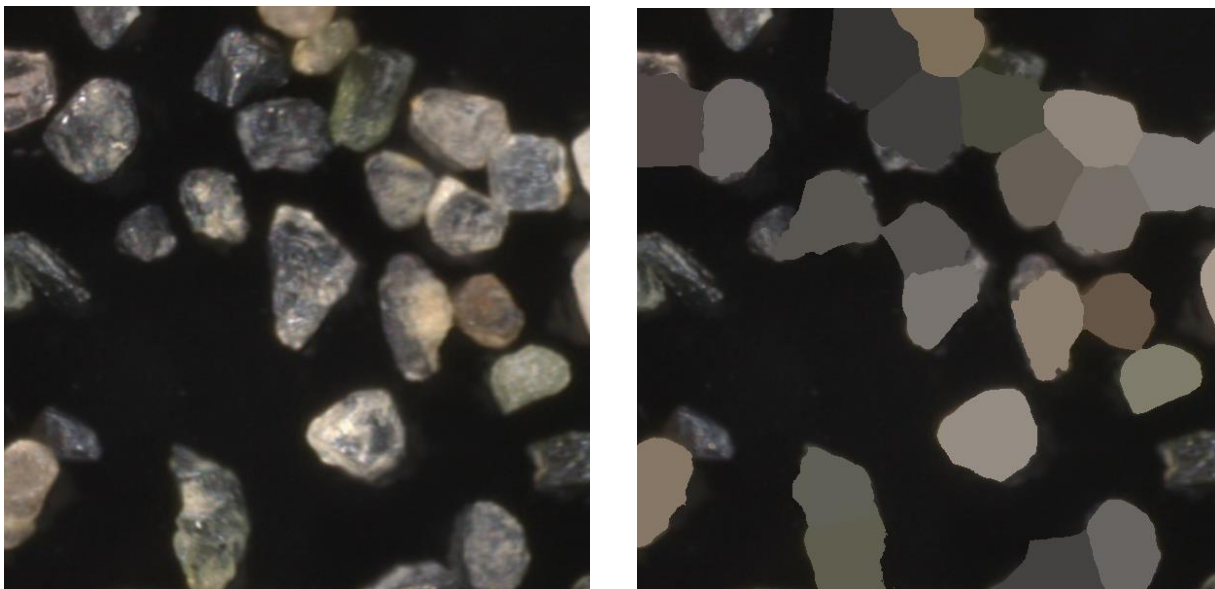
## Résultats obtenus

a. Image avec les différents contours trouvés



On constate que les segments trouvés ne sont pas aussi précis que ceux de l'algorithme des contours, cependant cet algorithme performe **nettement mieux** avec les grains qui se chevauchent. C'est la raison principale pour laquelle cet algorithme a été choisi.

b. Image avec les différentes couleurs remplies



Les grains qui ont une couleur sombre ne sont pas identifiés mais les grains qui se chevauchent sont repérés avec une assez bonne précision. Les couleurs moyennes identifiées correspondent assez bien aux couleurs des grains.



## Extraction de couleur

```
Test > ■ Image_SLIC.csv
1 Moyenne de B,Moyenne de G,Moyenne de R
2 81.52936188255246,83.874708501166,86.82213271146915
3 52.6150019636078,54.19046995680063,55.95902605053018
4 90.34021263289556,112.64915572232646,126.67667292057536
5 68.29363150218765,71.45486955112624,78.57316480311133
6 100.47537151444314,103.82668225079313,108.27717482050426
7 63.329984211281754,64.25520310033012,65.65465767188172
8 62.12368839427663,75.1737678855326,75.7181240063593
9 123.01754673663497,133.7140045916694,144.26221712036732
10 80.0953222954509,85.71628355569844,89.89455071531908
11 87.50810989738497,96.79510095994704,103.26216484607745
12 103.44989511053735,111.76488623527513,118.93916411166694
13 117.67204385277995,122.67768206734534,127.33876272513704
14 112.67495819397993,116.81103678929766,121.93561872909699
15 141.45561139028476,155.81825795644892,168.23283082077052
16 110.3343937787204,126.47384234711912,139.24779073877696
17 71.32479964381122,86.42764915405165,101.99599287622439
18 132.5443269908386,142.6982381959126,150.11571529245947
19 108.42807262569832,125.86336126629423,128.7709497206704
20 103.32822665267577,119.24176285414481,134.28709338929696
21 87.9284309870232,96.00117970900511,97.9431773495871
22 79.91706775268419,94.21269900037024,95.78822658274713
23 65.99210006583279,67.2238314680711,69.94930875576037
24 98.92694983522519,101.63218601244965,105.26931526913219
25 |
```

23 grains sont identifiés, 4 de plus que l'algorithme précédent car l'algorithme SLIC ne groupe pas les grains qui se chevauchent en un seul grain.

Le fait d'avoir les couleurs moyennes affichées sur l'image résultat permet de vérifier que le calcul de moyenne des segments se fait correctement.



## Exécution du code final

### 1. Exécution du code

```
PS C:\Users\paule\OneDrive\Paul-Emmanuel\UQAC 1\8INF804 - Vision Artificielle\TP2_ATTE BERTRAND EVORA TOUGMA> ls

Directory: C:\Users\paule\OneDrive\Paul-Emmanuel\UQAC 1\8INF804 - Vision Artificielle\TP2_ATTE BERTRAND EVORA TOUGMA

Mode                LastWriteTime         Length Name
----                -
da---l            08-Nov-23   05:48 PM             Images
d---l            08-Nov-23   05:45 PM             Rapport
-a---l            08-Nov-23   05:46 PM              1143 run.py
-a---l            08-Nov-23   05:26 PM              2822 SLIC.py

PS C:\Users\paule\OneDrive\Paul-Emmanuel\UQAC 1\8INF804 - Vision Artificielle\TP2_ATTE BERTRAND EVORA TOUGMA> python run.py Images
Processing file: Images\Echantillon1Mod2_301.png
Processing file: Images\Echantillon1Mod2_302.png
Processing file: Images\Echantillon1Mod2_303.png
Processing file: Images\Echantillon1Mod2_304.png
Processing file: Images\Echantillon1Mod2_305.png
Processing file: Images\Echantillon1Mod2_306.png
Processing file: Images\Echantillon1Mod2_316.png
Processing file: Images\Echantillon1Mod2_422.png
Processing file: Images\Echantillon1Mod2_471.png

PS C:\Users\paule\OneDrive\Paul-Emmanuel\UQAC 1\8INF804 - Vision Artificielle\TP2_ATTE BERTRAND EVORA TOUGMA\Images> ls

Directory: C:\Users\paule\OneDrive\Paul-Emmanuel\UQAC 1\8INF804 - Vision Artificielle\TP2_ATTE BERTRAND EVORA TOUGMA\Images

Mode                LastWriteTime         Length Name
----                -
-a---l            12-May-20   08:58 PM      476328 Echantillon1Mod2_301.png
-a---l            08-Nov-23   05:49 PM      221718 Echantillon1Mod2_301z_with_SLIC.png
-a---l            12-May-20   08:58 PM      522917 Echantillon1Mod2_302.png
-a---l            08-Nov-23   05:49 PM      213012 Echantillon1Mod2_302z_with_SLIC.png
-a---l            12-May-20   08:58 PM      522473 Echantillon1Mod2_303.png
-a---l            08-Nov-23   05:49 PM      227163 Echantillon1Mod2_303z_with_SLIC.png
-a---l            12-May-20   08:58 PM      501473 Echantillon1Mod2_304.png
-a---l            08-Nov-23   05:49 PM      232887 Echantillon1Mod2_304z_with_SLIC.png
-a---l            12-May-20   08:58 PM      490674 Echantillon1Mod2_305.png
-a---l            08-Nov-23   05:49 PM      230959 Echantillon1Mod2_305z_with_SLIC.png
-a---l            12-May-20   08:58 PM      477226 Echantillon1Mod2_306.png
-a---l            08-Nov-23   05:49 PM      222506 Echantillon1Mod2_306z_with_SLIC.png
-a---l            12-May-20   08:58 PM      397743 Echantillon1Mod2_316.png
-a---l            08-Nov-23   05:49 PM      220475 Echantillon1Mod2_316z_with_SLIC.png
-a---l            12-May-20   08:58 PM      430011 Echantillon1Mod2_422.png
-a---l            08-Nov-23   05:49 PM      225010 Echantillon1Mod2_422z_with_SLIC.png
-a---l            12-May-20   08:58 PM      504438 Echantillon1Mod2_471.png
-a---l            08-Nov-23   05:49 PM      168524 Echantillon1Mod2_471z_with_SLIC.png
-a---l            08-Nov-23   05:49 PM       15144 image.csv
```

On constate que les images traitées ont été créées et le fichier csv est bien créé. Nous préférons garder l'image originelle pour mieux percevoir la segmentation. On compare l'image originelle avec l'image segmentée.

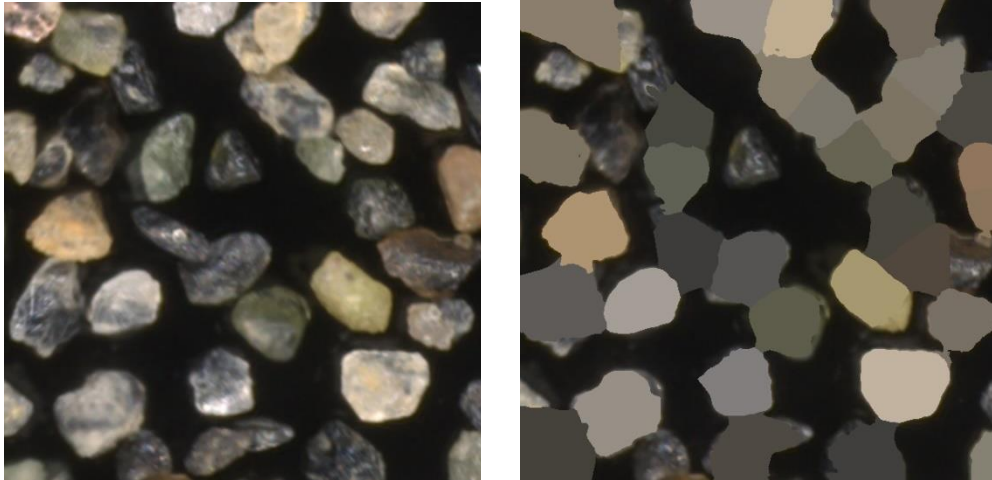
### 2. Dataframe final

```
elp
SLIC.py  image.csv x run.py
Images > image.csv
1 Moyenne de B,Moyenne de G,Moyenne de R
2 60.06812609457093,73.98388791593695,79.24133099824869
3 143.89450387310956,171.16506824050165,196.00903725562523
4 68.21844985481349,80.19186955550592,88.31963368327004
5 102.21006796941376,116.56393372982158,122.12022090059473
6 90.00177619893428,108.8595026642984,109.8202486678508
7 81.36439393939393,81.30176767676768,83.52234848484848
8 133.28762949421085,144.49329677026205,151.7318708104814
9 168.19702689948087,186.3957055214724,199.74846625766872
10 109.3639942734431,132.69452397995704,151.2219040801718
11 110.3133263743911,141.79418928322805,113.89213639526793
```

Le dataframe final est créé et il contient 270 grains qui ont été identifiés dans les images du dossier. Les grains sombres n'ont pas été identifiés et certains grains ayant une coloration particulière sont identifiés comme deux grains. Néanmoins le résultat final obtenu nous semble correcte.

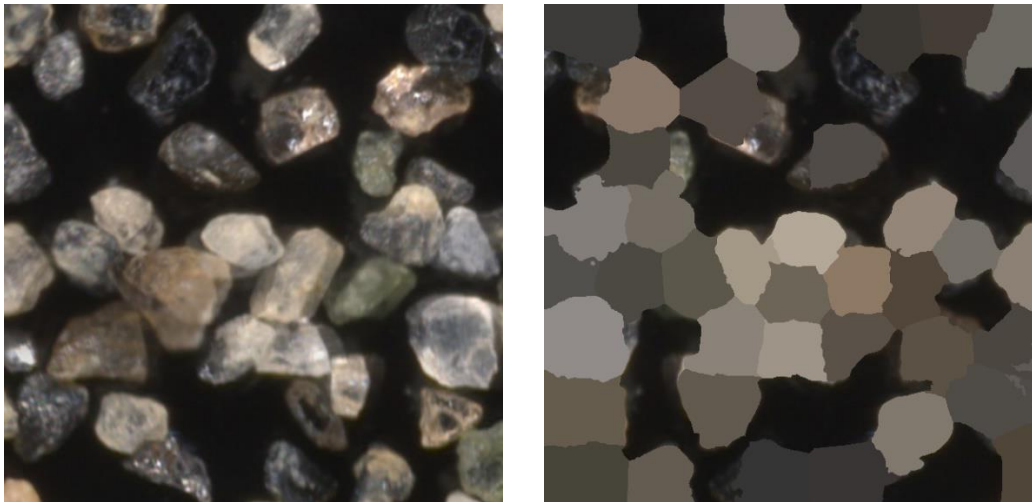
ATTE, BERTRAND, EVORA, TOUGMA

### Bon résultat



On constate que pour cette image la segmentation s'est bien effectuée avec les couleurs moyennes qui correspondent bien aux grains.

### Résultat moyen



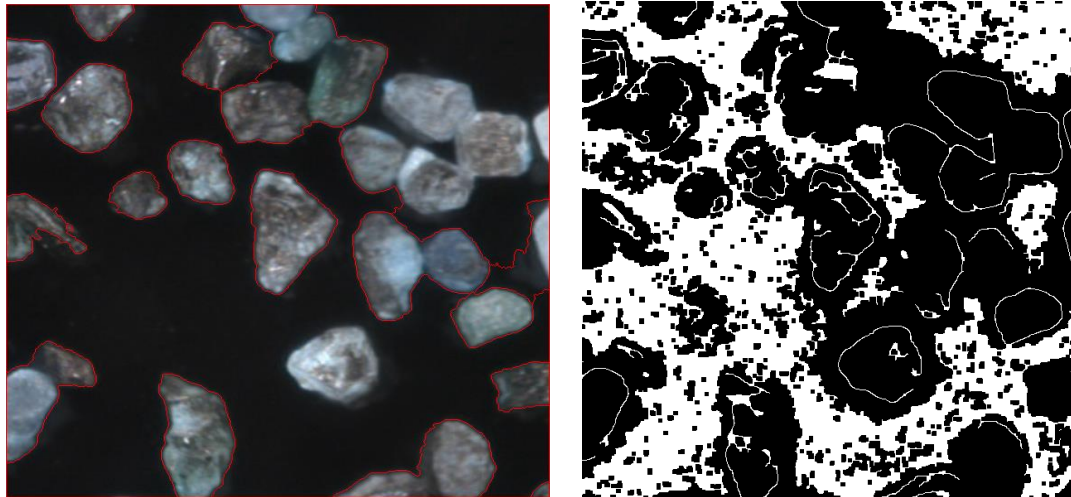
La segmentation est moins bonne pour cette image car l'image originelle contient des faux reflets, un prétraitement spécifique devrait être appliqué pour avoir un meilleur résultat. Cependant nous évitons cela afin de ne pas faire du « overfitting » de l'algorithme.

### Difficultés rencontrées

La difficulté principale de ce travail était de s'informer sur les différentes méthodes de segmentation et les méthodes de segmentation qui vont avec.

Le travail a commencé avec une grosse partie de « trial and error » où nous testons différentes méthodes et en fonction des résultats obtenus nous jugeons leur pertinence pour ce problème.

### Segmentation Watershed (gauche) et Canny edge detector (droite)



Par exemple la méthode Watershed semblait avoir plus de mal que la méthode « Contour based » avec les **grains qui étaient regroupés**. Tandis que le Canny edge detector ne trouvait pas assez de contours fermés ainsi l'opération Fill donnait des résultats **insatisfaisants**.

Parmi les premières méthodes testées, la segmentation à l'aide des contours donnait les meilleurs résultats mais le chevauchement était un problème principal ainsi la segmentation SLIC a été choisi.

### Conclusion

En conclusion lors de ce TP nous appliquons différentes techniques de segmentation afin de pouvoir distinguer les grains dans une image. A la suite de cette segmentation nous extrayons la couleur moyenne de chaque segment. Différentes méthodes ont été explorées lors de ce travail mais la technique qui donnait les résultats les plus intéressants pour l'extraction de couleur était la segmentation SLIC (Simple Linear Iterative Clustering).

Nous pouvons donc exploiter le dataframe généré à la fin pour entrainer un modèle de clustering qui pourrait grouper les grains semblables en fonction de leur couleur. Des features supplémentaires tel que l'aire du grain pourrait être ajoutées au dataframe pour rendre le modèle plus performant.