TPs de Vision

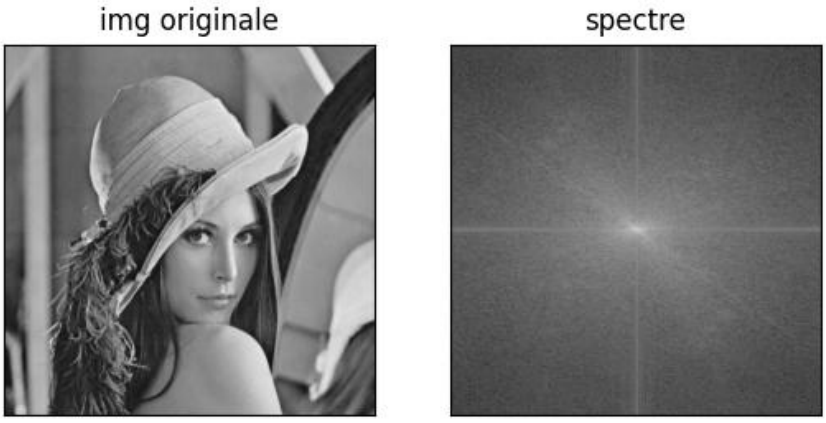
Elio Genson

Table des matières

TP1 - Caractérisation fréquentielle de la texture

I.Filtrage fréquentiel

1. Image du spectre de Fourier :



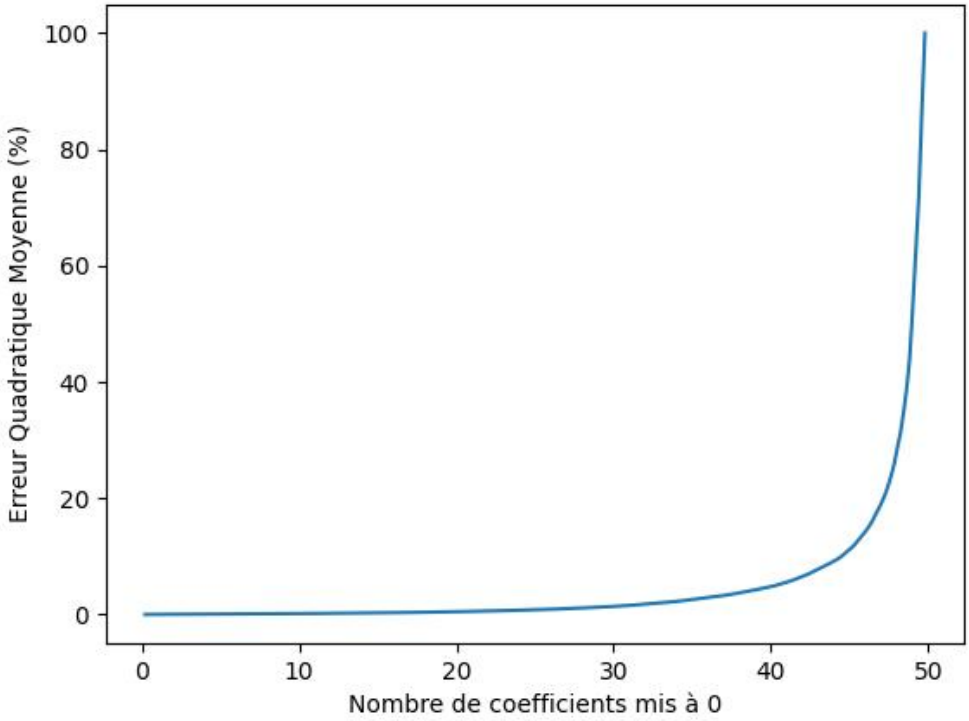
2. Image reconstruite :

Le MSE calculé entre les deux images est de 17553

3. Images avec coefficient à 0 :

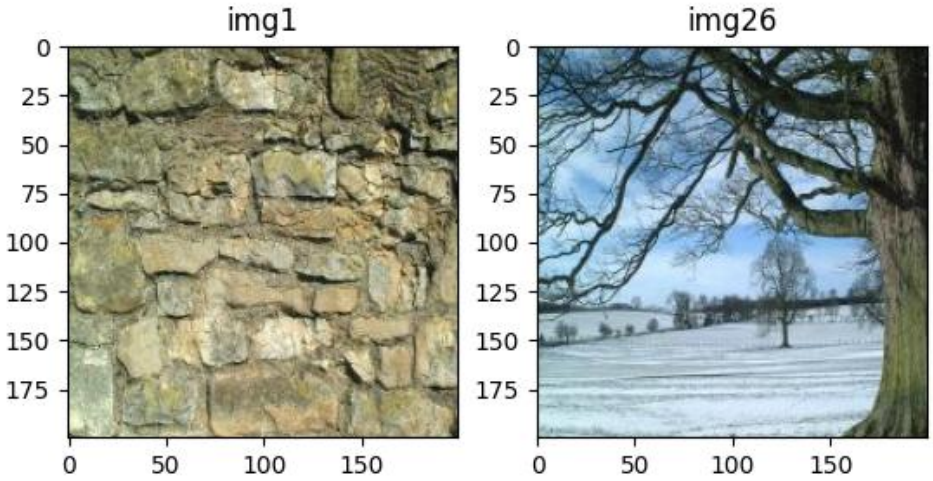
On remarque que l'image n=200 est bien détérioré, il y a des «vagues» sur les contours des formes.

4. Graphe de la MSE en fonction du nombre de coef mit à zero :



II.Caractérisation

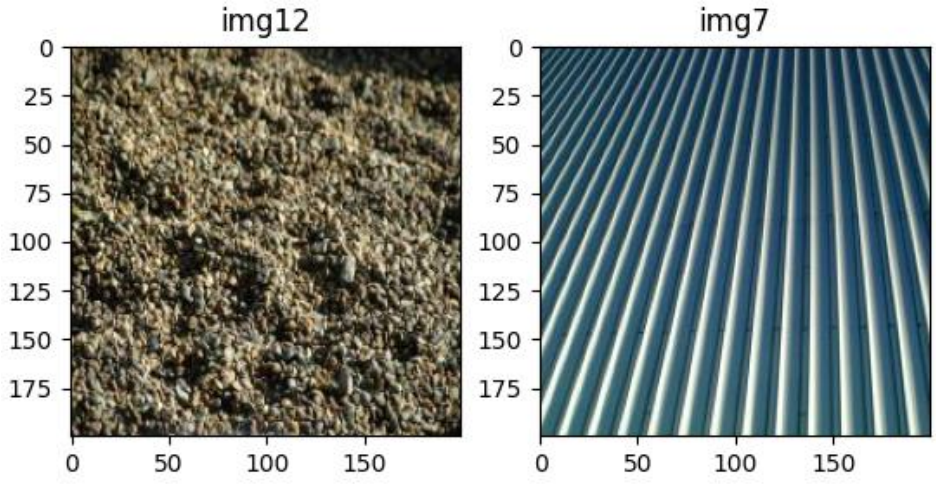
Pour l'image 1 :
Distance de manhantan min = 249



Autres testes :

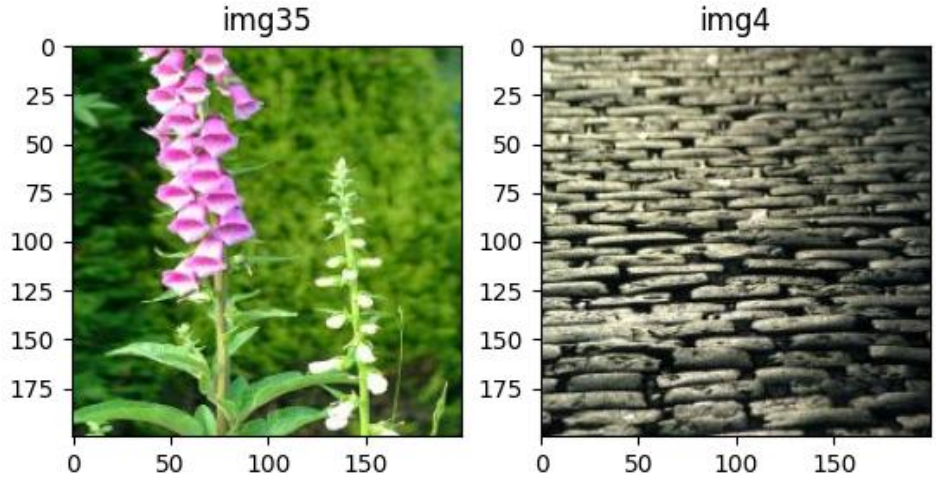
Img 12

Distance min = 35913

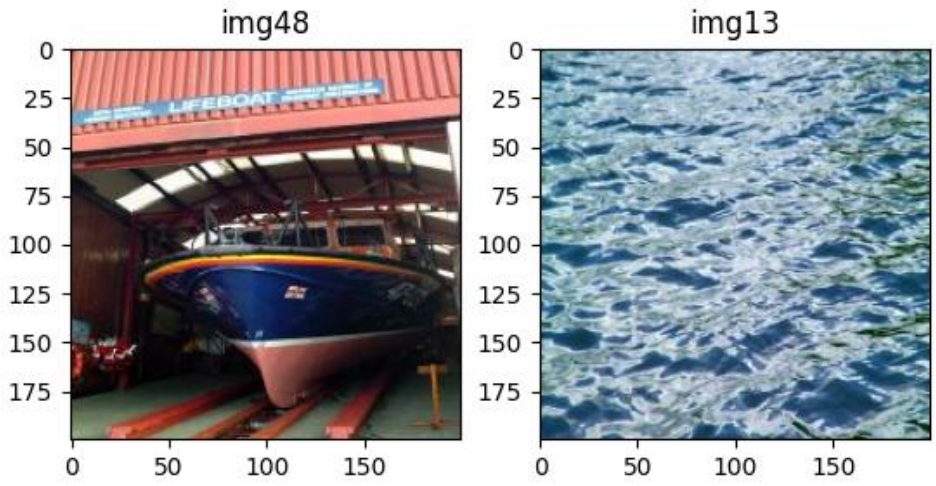


Img 35

Distance min = 9658



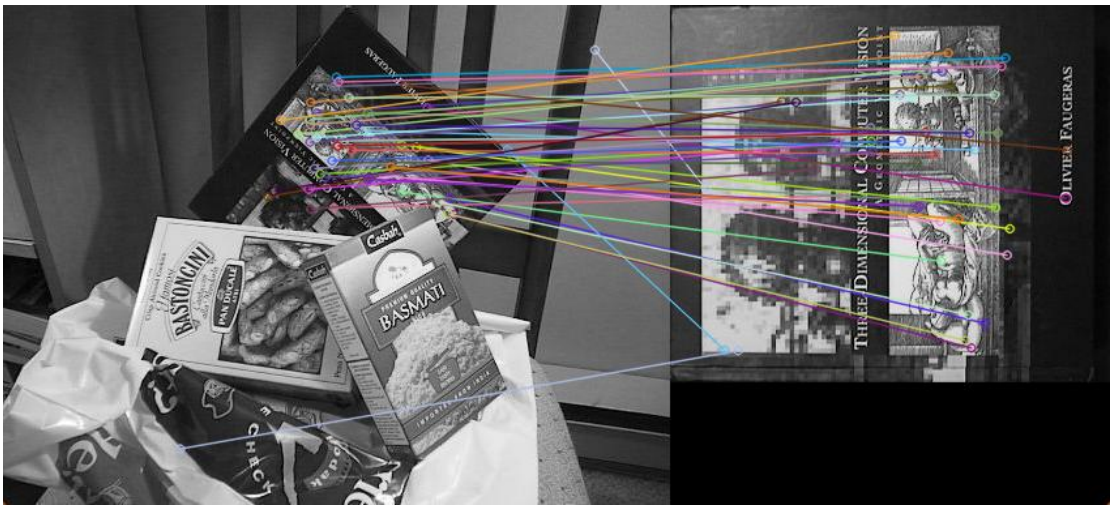
Img 48
Distance min = 8385



Je ne retrouve pas beaucoup de corrélation au niveaux des textures des images testés.
les seuls images qui on un rapport son la 35 et la 4 ou les feuilles on le même style de texture que les pavets.

TP2 - Points d'intérêt

I. SIFTs



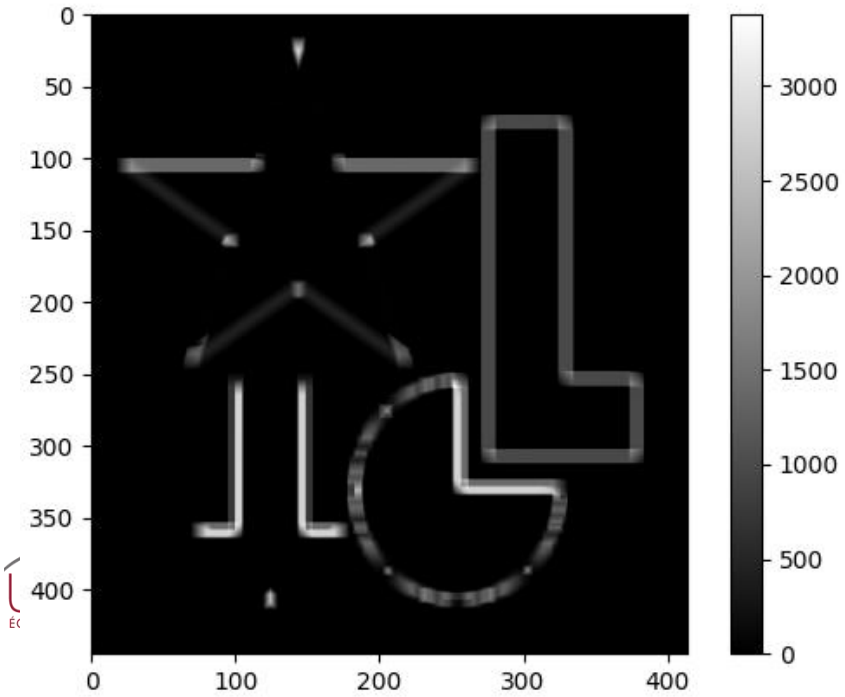
Description du code :

cv.SIFT_create() crée un détecteur SIFT.
detectAndCompute(img, None) détecte les points d'intérêt (keypoints) et calcule leurs descripteurs.
k_1, k_2 : listes des points clés détectés.
des_1, des_2 : matrices contenant les descripteurs SIFT des points clés.

cv.BFMatcher (Brute-Force Matcher) est un algorithme d'appariement descripteur par descripteur.
bf.match(des_1, des_2) apparie les descripteurs des deux images.
sorted(matches, key=lambda x: x.distance) trie les correspondances par distance croissante (les meilleures correspondances en premier).

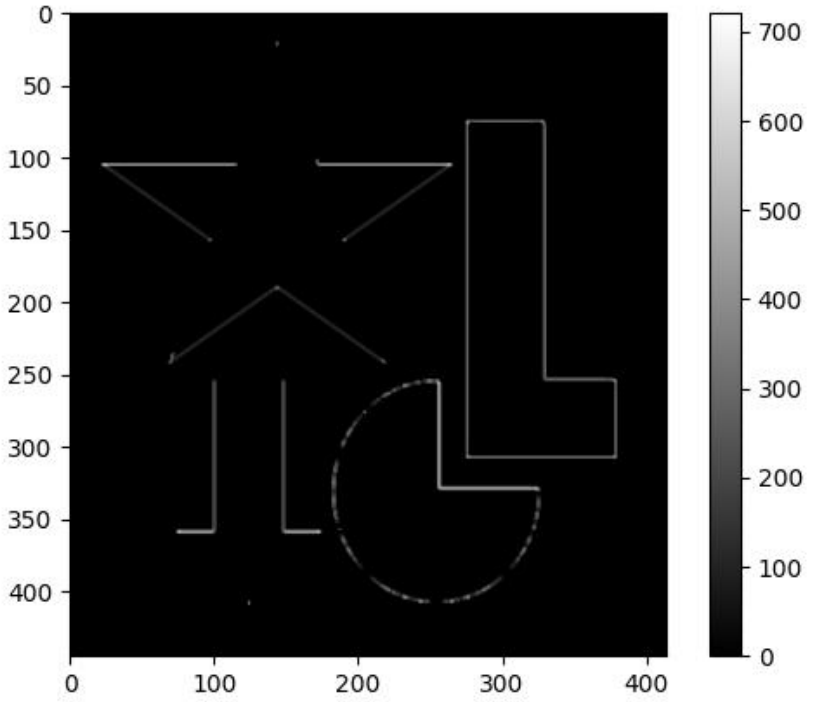
II. Points de Moravec :

Fenetre de taille 11



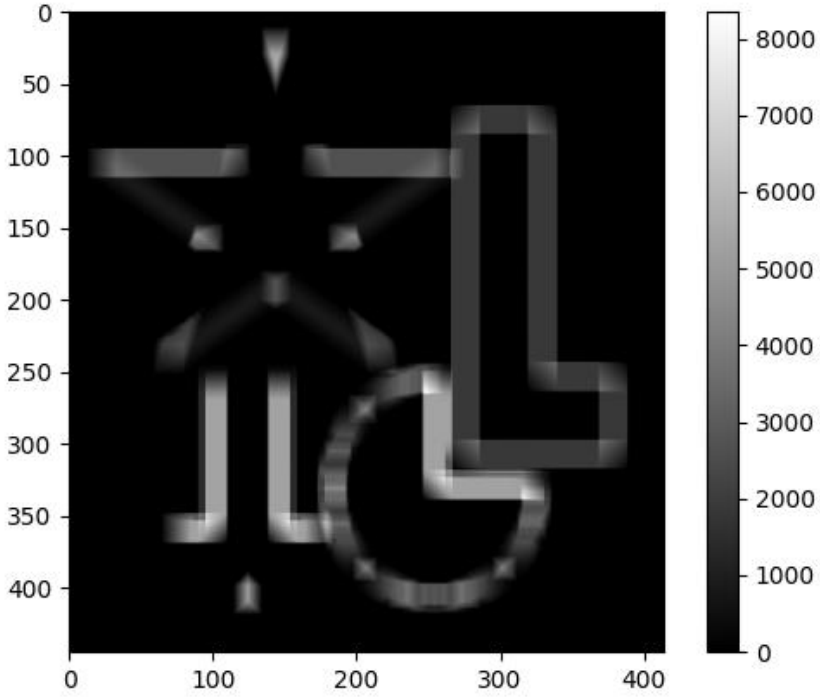
43 points d'intéret

fenêtre de taille 3



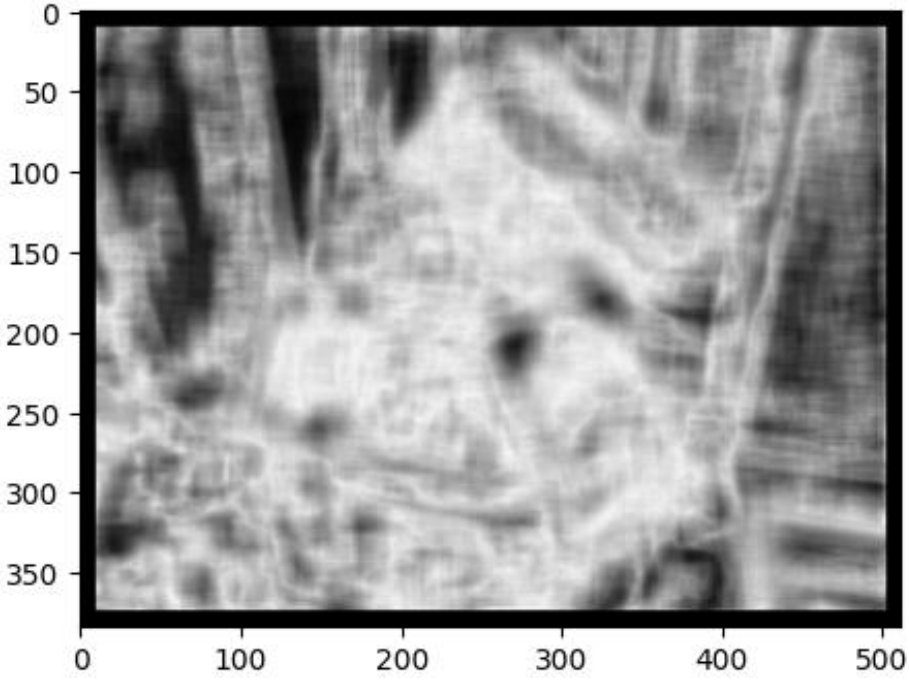
233 points d'intéret

fenêtre de taille 21



233 points d'intéret

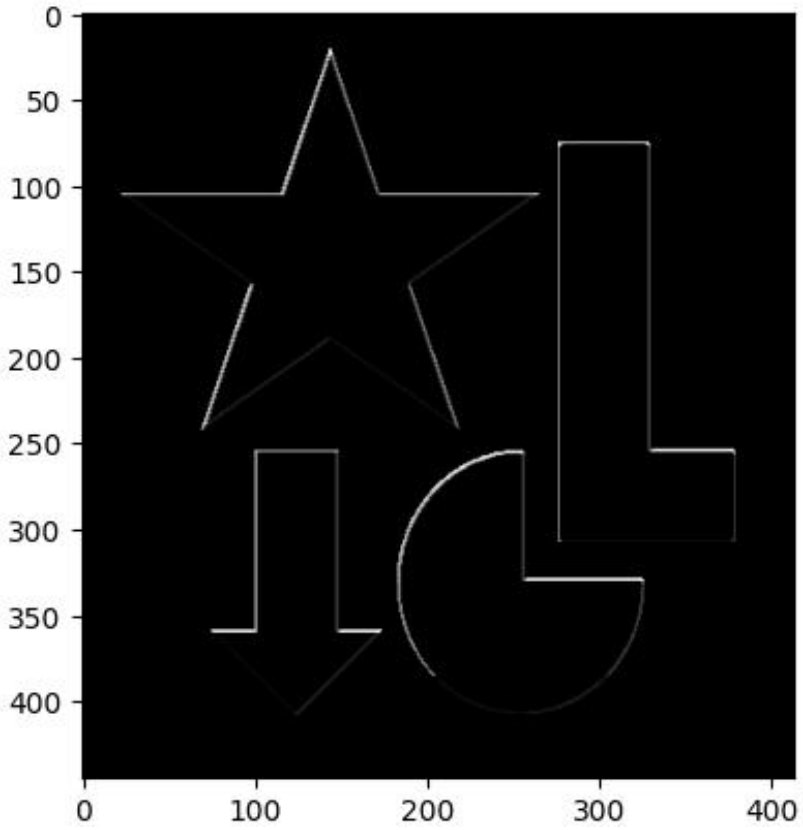
sur l'image dela partie 1 en 11 pixels :



8603 points de moravec

plus le nombre de pixel de voisinage est grand, plus nombreux sont les point de moravec détecté

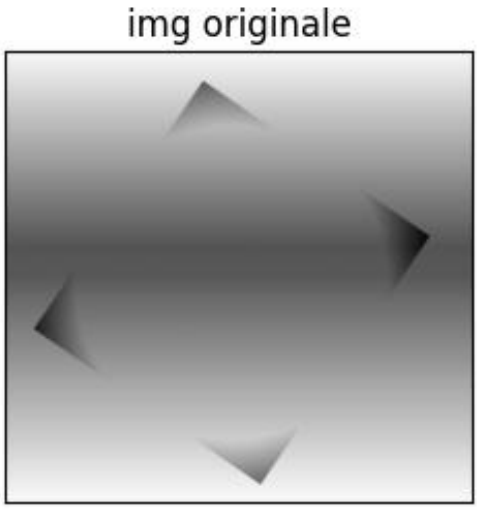
III. LBP



11 point d'intéret

TP3 - Transformée de Hough

I. Modélisation de segments de droite.



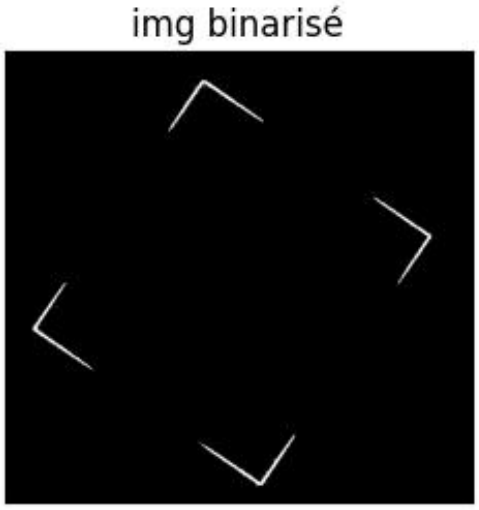
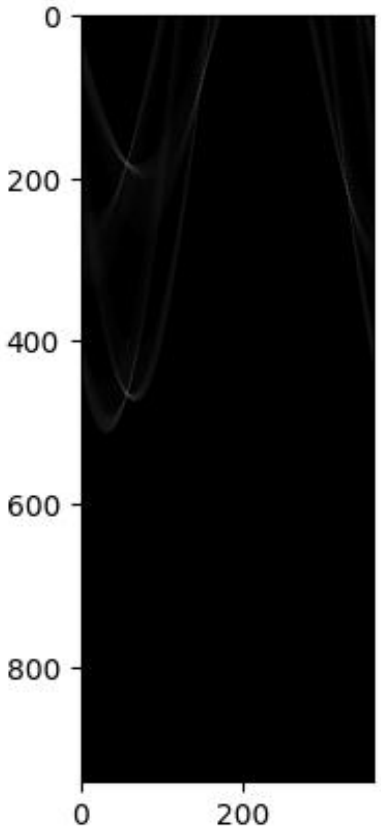
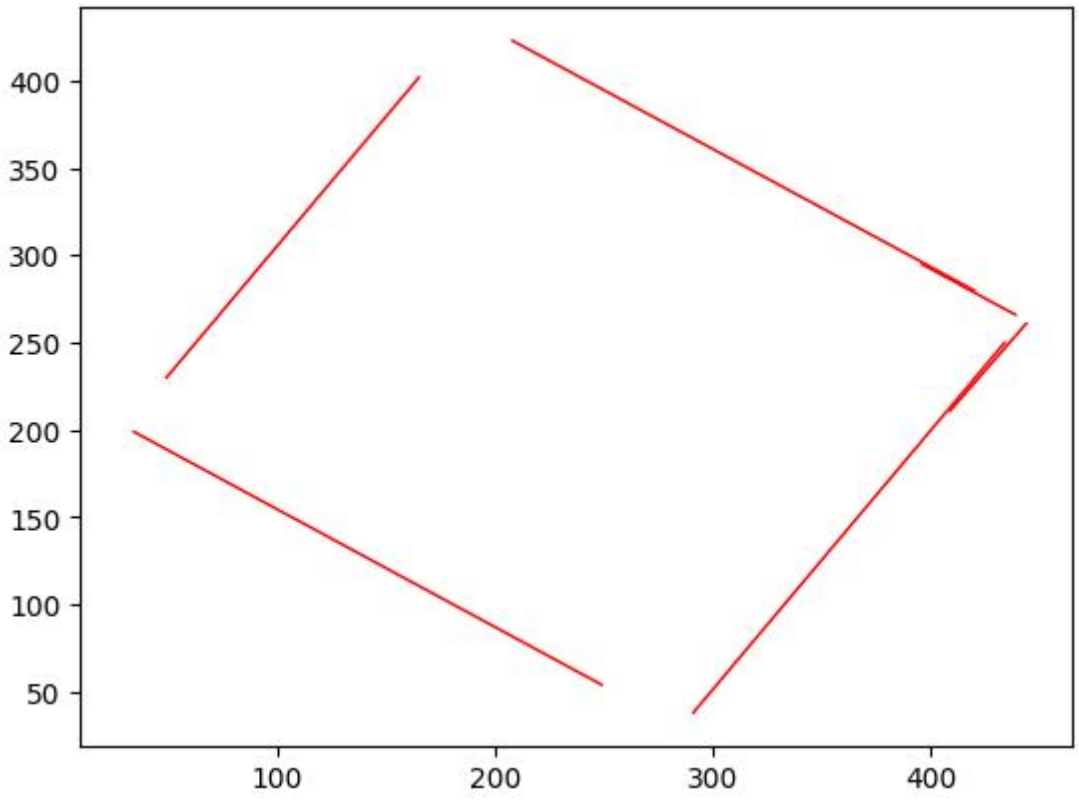


image de Vote :



II. Reconnaissance de segments de droite



k=10

Point 10 : 467 56 (208 , 423) (420 , 280)

Point 9 : 183 56 (265 , 42) (265 , 42)

Point 8 : 93 145 (30 , 205) (30 , 205)

Point 7 : 184 56 (34 , 199) (249 , 54)

Point 6 : 219 326 (308 , 65) (308 , 65)

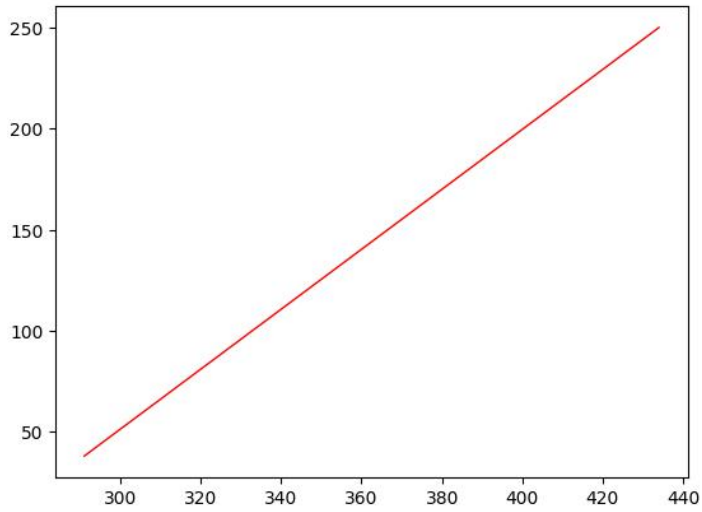
Point 5 : 466 56 (396 , 295) (439 , 266)

Point 4 : 214 325 (409 , 211) (444 , 261)

Point 3 : 88 146 (49 , 230) (165 , 402)

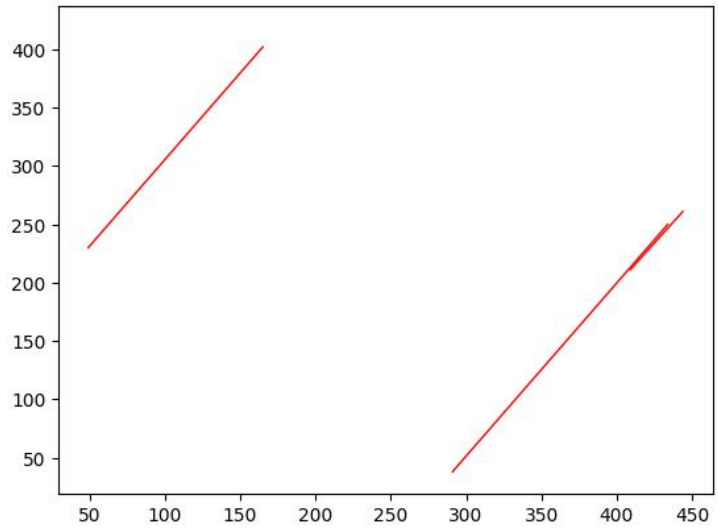
Point 2 : 87 146 (177 , 418) (177 , 418)

Point 1 : 220 326 (291 , 38) (434 , 250)



K=1

Point 1 : 220 326 (291 , 38) (434 , 250)

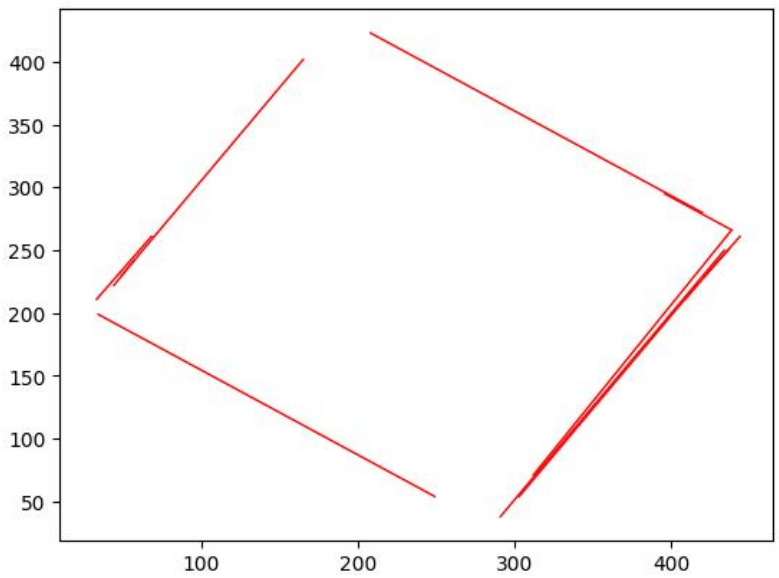


K=4

Point 4 : 214 325 (409 , 211) (444 , 261)

Point 3 : 88 146 (49 , 230) (165 , 402)

Point 2 : 87 146 (177 , 418) (177 , 418)



Point 1 : 220 326 (291 , 38) (434 , 250)

K= 20

Point 20 : 228 327 (442 , 262) (442 , 262)
Point 19 : 206 324 (421 , 229) (437 , 251)
Point 17 : 84 147 (44 , 222) (57 , 242)
Point 16 : 223 327 (312 , 71) (438 , 265)
Point 15 : 94 145 (33 , 211) (68 , 261)
Point 13 : 215 325 (413 , 215) (413 , 215)
Point 11 : 221 326 (303 , 54) (417 , 223)
Point 10 : 467 56 (208 , 423) (420 , 280)
Point 9 : 183 56 (265 , 42) (265 , 42)
Point 8 : 93 145 (30 , 205) (30 , 205)
Point 7 : 184 56 (34 , 199) (249 , 54)
Point 6 : 219 326 (308 , 65) (308 , 65)
Point 5 : 466 56 (396 , 295) (439 , 266)
Point 4 : 214 325 (409 , 211) (444 , 261)
Point 3 : 88 146 (49 , 230) (165 , 402)
Point 2 : 87 146 (177 , 418) (177 , 418)
Point 1 : 220 326 (291 , 38) (434 , 250)

III. Rétroprojection de la transformée de Radon

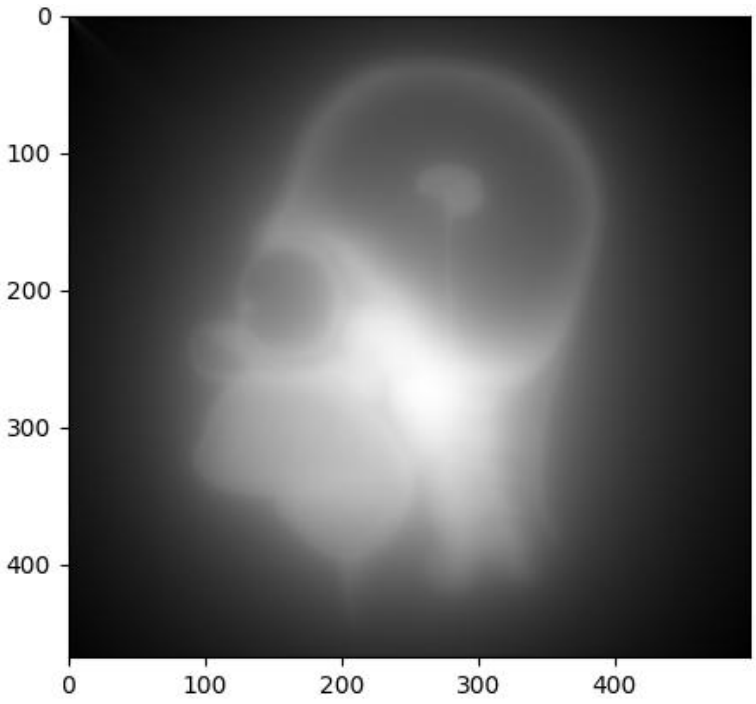
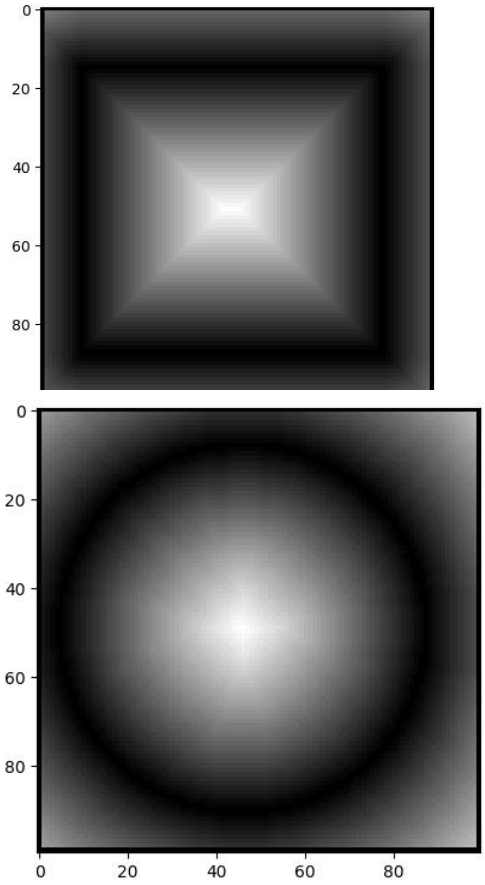


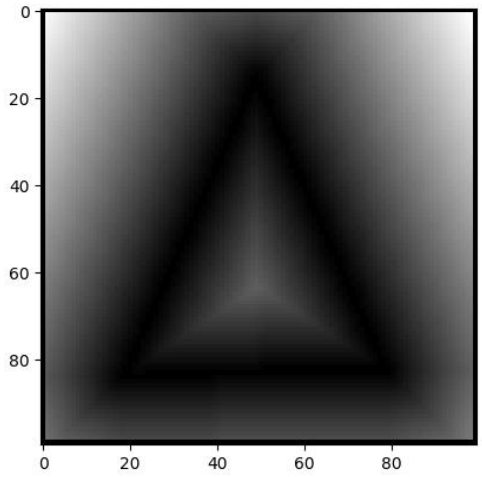
image de radon reconstruite

l'organe est un cerveau !

TP4 - Transformée en distance.

I. Modélisation





Transformées en distance calculées avec ce masque :

[[4. 3. 4.]
[3. 0. 3.]
[4. 3. 4.]]

II.Reconnaissance

Reco en cercle

Reco en cercle

reco en Triangle

Une erreur avec le carré reco en cercle.

carre: {'triangle2': 66577497.0, 'carre2': 69139059.0, 'cercle2': 69553092.0}
triangle: {'cercle2': 101068827.0, 'carre2': 101641816.0, 'triangle2': 105029518.0}
cercle: {'triangle2': 76482353.0, 'carre2': 78693061.0, 'cercle2': 79128188.0}
Score le plus haut = forme reconnu

TP5 - Forêt aléatoire

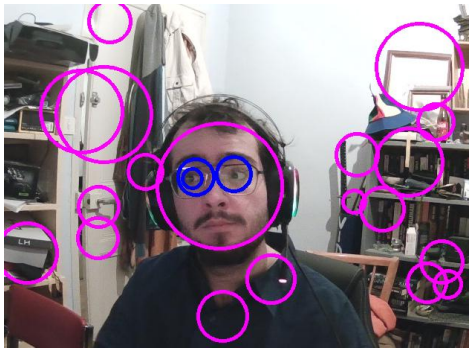
Je calcule 3301 haar features sur les images avec un bloc 10x10 pixels

Je n’ai pas réussi le restes des exercices de ce TP

TP6 - Cascades de Haar et Yolo

Teste sur un visage :

Teste visage + yeux :



On remarque beaucoup d'erreur de détection sur un fond chargé.

Détection de corps entier:

Elle aussi n'est pas très précise.

Précision des cascades :


```
{
    "body + faces": {
        "positif": 261,
        "negatif": 121,
        "accuracy": 68.32460732984293,
        "recall": 52.2
    },
    "faces only": {
        "positif": 195,
        "negatif": 64,
        "accuracy": 75.2895752895753,
        "recall": 39.0
    },
```

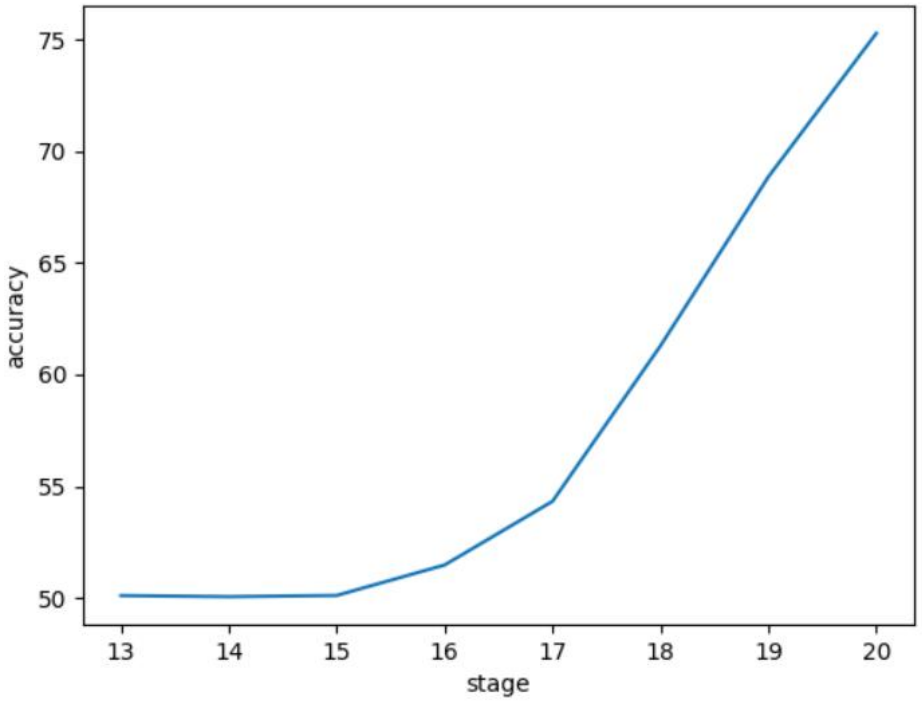

La cascade du visage à 20 niveaux de 9 classifieurs faibles chacun.

Évolution de l'acuracy par niveaux supprimé :


```
    "faces whithout stage 19": {
        "positif":230,
        "negatif": 104,
        "accuracy": 68.8622754491018,
        "recall": 46.0
    },
    "faces whithout stage 18": {
        "positif": 285,
        "negatif": 180,
        "accuracy": 61.29032258064516,
        "recall": 56.99999999999999
    },
    "faces whithout stage 17": {
        "positif": 358,
        "negatif": 301,
        "accuracy": 54.3247344461305,
        "recall": 71.6
    },
    "faces whithout stage 16": {
        "positif": 420,
        "negatif": 396,
        "accuracy": 51.470588235294116,
        "recall": 84.0
    },
```

```
    "faces whithout stage 15": {
        "positif": 471,
        "negatif": 469,
        "accuracy": 50.10638297872341,
        "recall": 94.19999999999999
    },
    "faces whithout stage 14": {
        "positif": 490,
        "negatif": 489,
        "accuracy": 50.05107252298263,
        "recall": 98.0
    },
    "faces whithout stage 13": {
        "positif": 497,
        "negatif": 495,
        "accuracy": 50.1008064516129,
        "recall": 99.4
    }
```


L'accuracy diminue quand on supprime les niveaux mais semble stagner à partie de la surpression du 15eme niveau. Les différent niveaux joue donc un rôle dans l'efficacité de la détection. L'accurary n'est pas une bonne mesure de performance seul, et est souvent couplé au recall dans le F1 score.



YOLO

Dans la description des classes, «person» se trouve sur la 1er ligne.

Sur la base de testes l'accuracy du modèle yolo est de 50.07 %