Exam TI / GPU

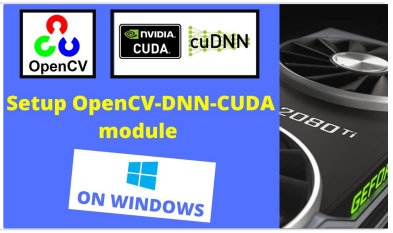
Cours | GPU + Traitement d'images

Installer la version d'OpenCV compilé avec le support de CUDA

Setup OpenCV-DNN module with CUDA backend support on Windows

This video shows step by step tutorial on how to set up the OpenCV-DNN module with CUDA backend support on Windows.

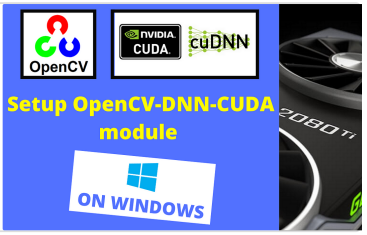
https://www.youtube.com/watch?v=VAoNMlCm_Cg



Setup OpenCV-DNN module with CUDA backend support (For Windows)

In this tutorial, we will be building OpenCV from source with CUDA backend support (OpenCV-DNN-CUDA module).

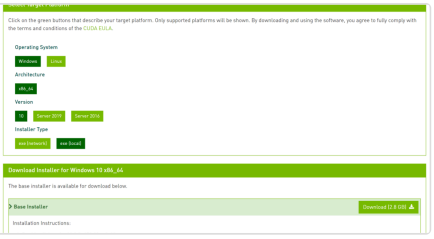
https://medium.com/geekculture/setup-opencv-dnn-module-with-cuda-backend-support-for-windows-7f1856691da3



Install CUDA and CUDNN on Windows & Linux

CONTENTS

https://medium.com/geekculture/install-cuda-and-cudnn-on-windows-linux-52d1501a8805



<u>Vérification si tout est bien installé</u>

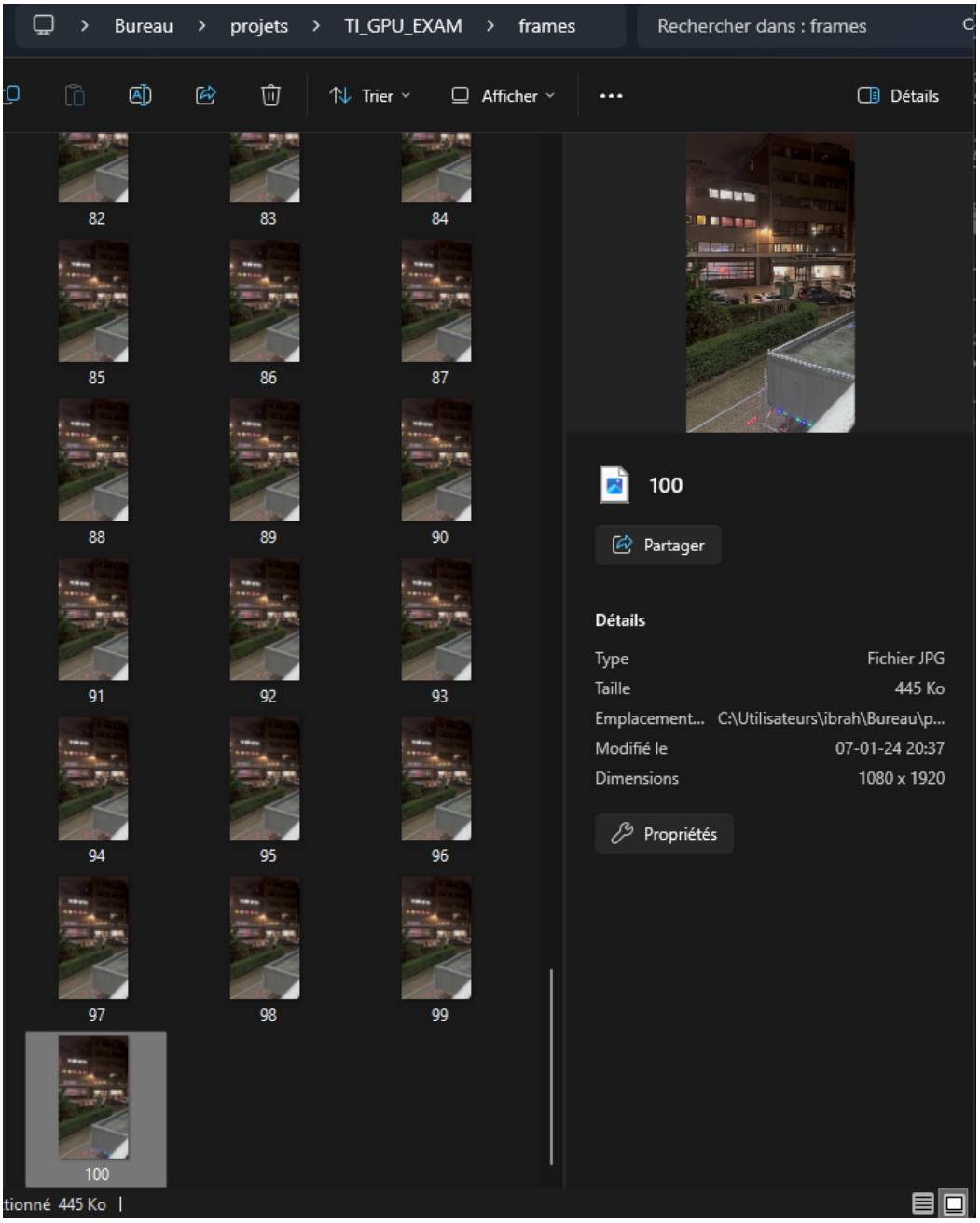
```
import numpy as np
import cv2 as cv
import time
```

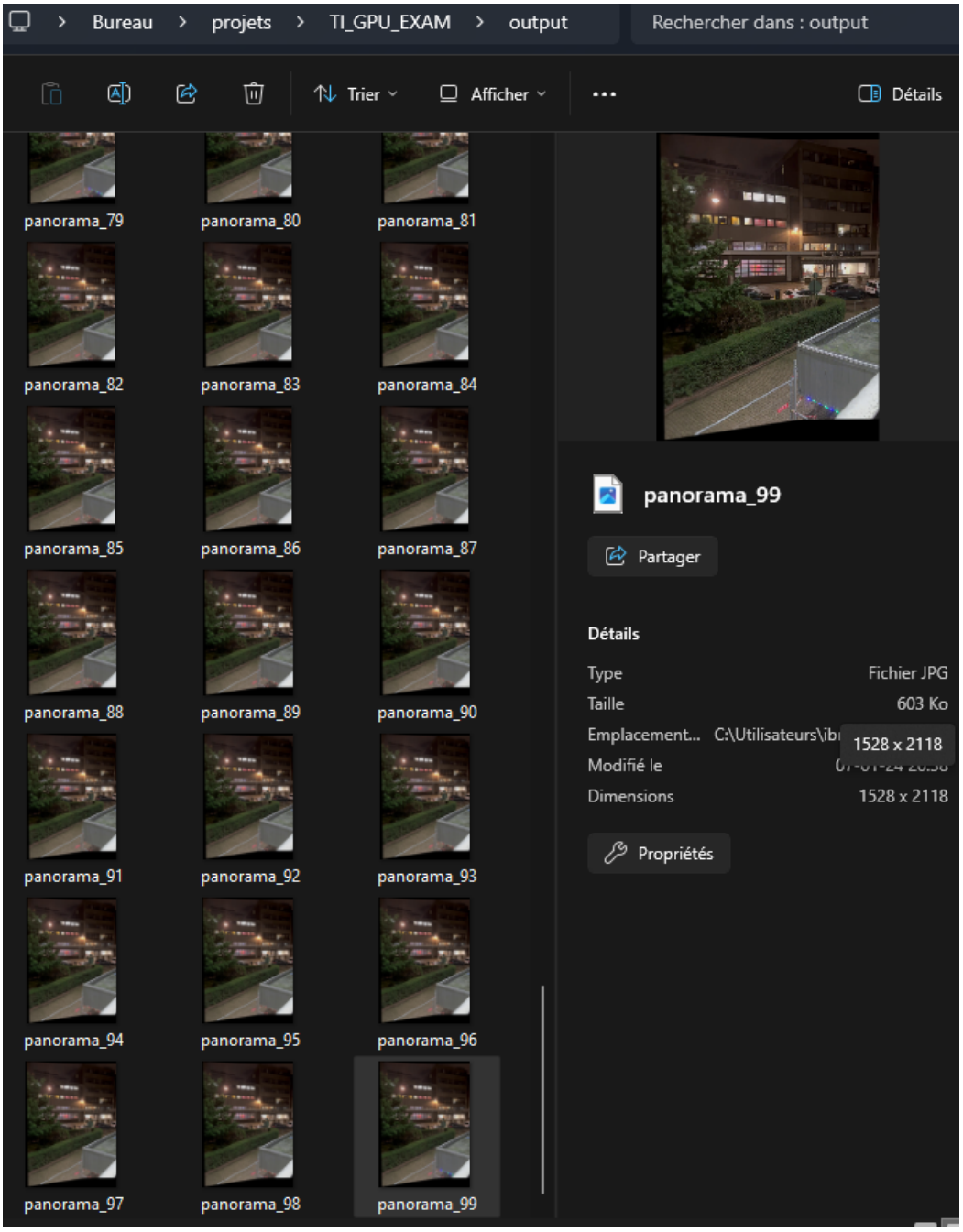
```
npTmp = np.random.random((1024, 1024)).astype(np.float32)
npMat1 = np.stack([npTmp,npTmp],axis=2)
npMat2 = npMat1
cuMat1 = cv.cuda_GpuMat()
cuMat2 = cv.cuda_GpuMat()
cuMat1.upload(npMat1)
cuMat2.upload(npMat2)
start_time = time.time()
cv.cuda.gemm(cuMat1, cuMat2,1,None,0,None,1)

print("CUDA using GPU --- %s seconds ---" % (time.time() - start_tim
start_time = time.time()
cv.gemm(npMat1,npMat2,1,None,0,None,1)
print("CPU --- %s seconds ---" % (time.time() - start_time))
```

output :

```
(env_gpu) PS C:\Users\ibrah\Desktop\projets\Traitement images> pytho
CUDA using GPU --- 0.11492133140563965 seconds ---
CPU --- 1.7428045272827148 seconds ---
```

==== ==Test pour 100 frames CPU =========


Bureau
projets
TI_GPU_EXAM
frames
Rechercher dans : frames
Trier
Afficher
Détails
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
100
Partager
Détails
Type
Fichier JPG
Taille
445 Ko
Emplacement...
C:\Utilisateurs\ibrah\Bureau\p...
Modifié le
07-01-24 20:37
Dimensions
1080 x 1920
Propriétés
tionné 445 Ko



Profilage de performance sans CUDA

The Python Profilers

Source code: Lib/profile.py and Lib/pstats.py Introduction to the profilers: cProfile and profile provide deterministic profiling of Python programs. A profile is a set of statistics that describes...

https://docs.python.org/3/library/profile.html

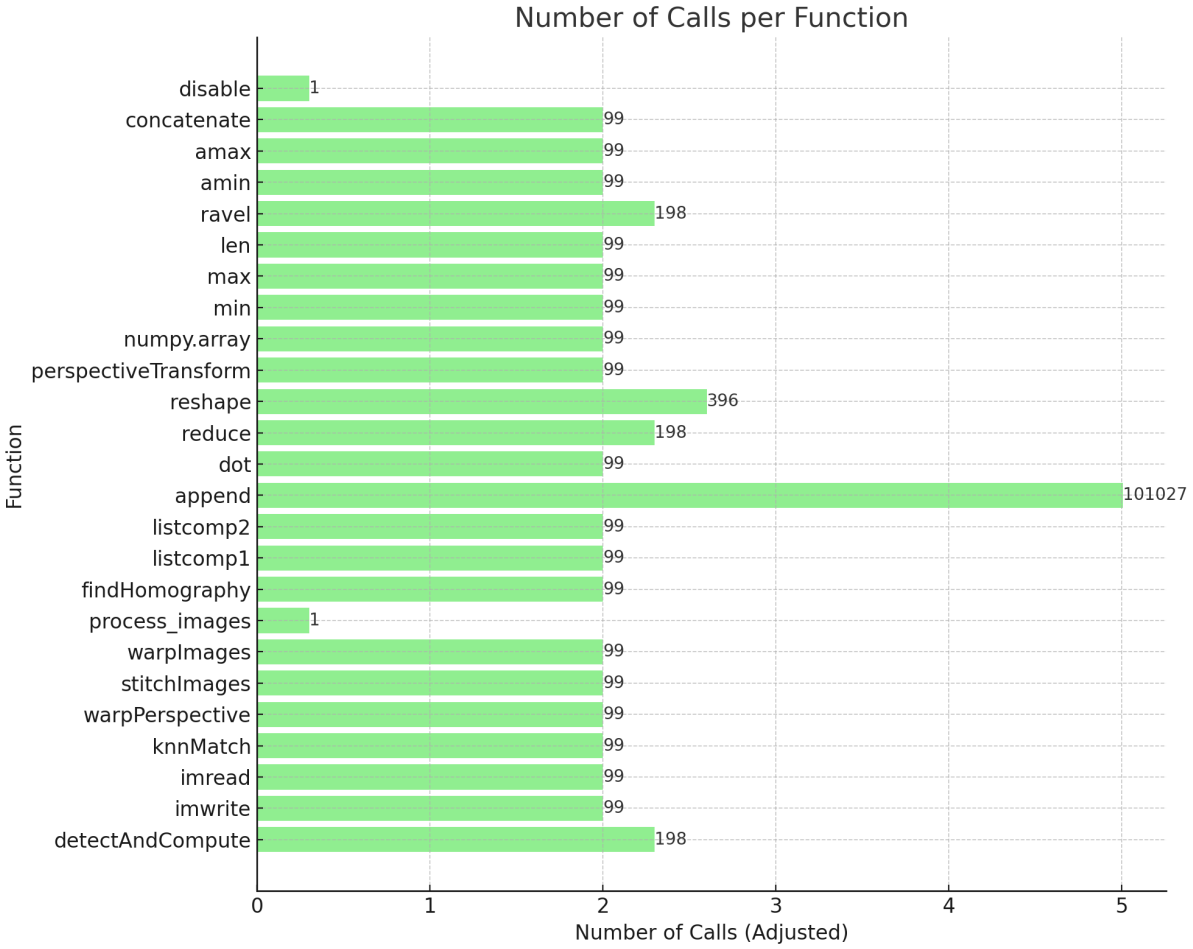
Pour commencer je fais un profilage de performance pour comprendre ou le programme passe plus de temps lors de son exécution, ce qui me permet d'identifier les parties de mon code pouvant être optimisé.

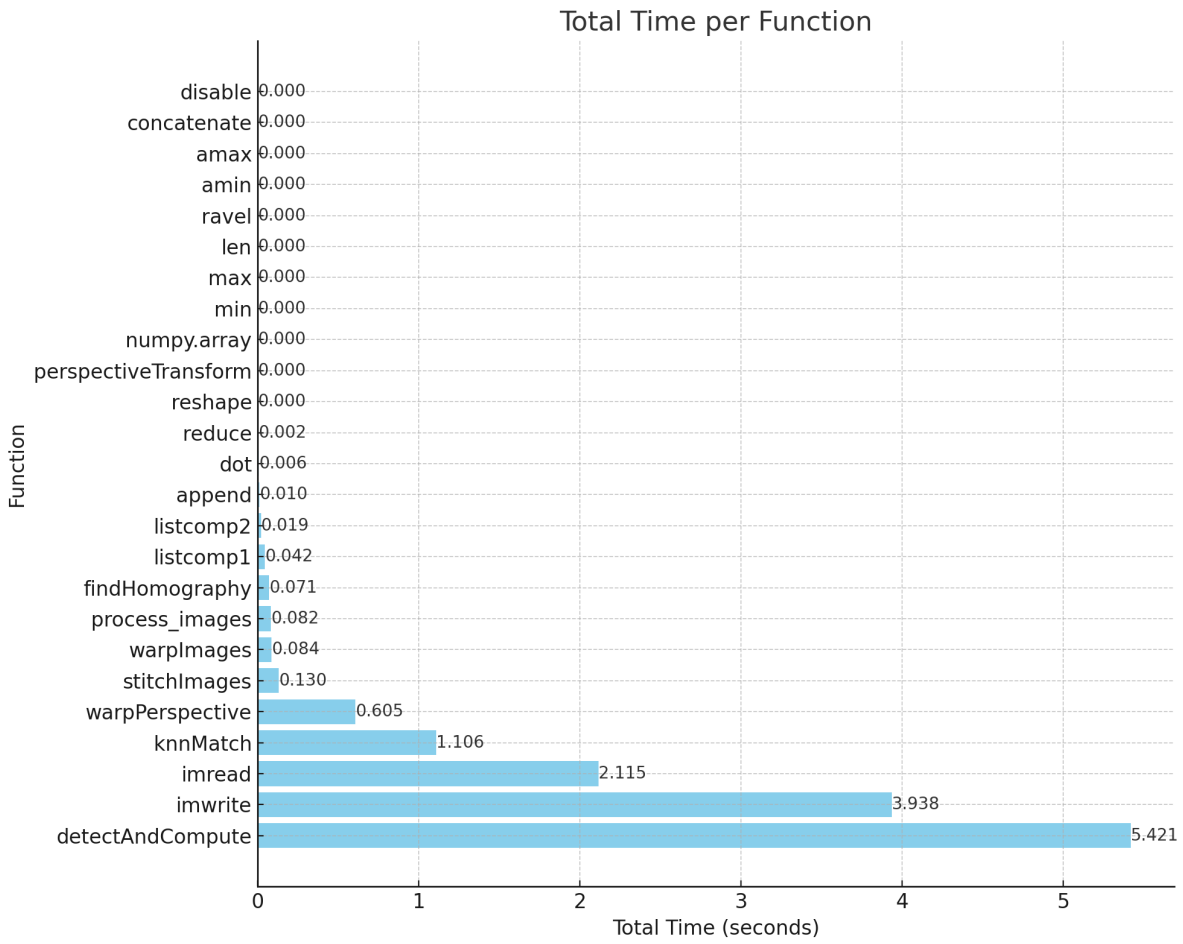
```
(env_gpu) PS C:\Users\ibrah\Desktop\projets\exam_Ti_gpu> python .\app.py
         103801 function calls in 13.633 seconds

   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      198    5.421    0.027    5.421    0.027 {method 'detectAndCompute' of 'cv2.Feature2D' objects}
       99    3.938    0.040    3.938    0.040 {imwrite}
       99    2.115    0.021    2.115    0.021 {imread}
       99    1.106    0.011    1.106    0.011 {method 'knnMatch' of 'cv2.DescriptorMatcher' objects}
       99    0.605    0.006    0.605    0.006 {warpPerspective}
       99    0.130    0.001    7.498    0.076 app.py:71(stitchImages)
       99    0.084    0.001    0.699    0.007 app.py:44(warpImages)
        1    0.082    0.082   13.633   13.633 app.py:96(process_images)
       99    0.071    0.001    0.071    0.001 {findHomography}
       99    0.042    0.000    0.042    0.000 app.py:84(<listcomp>)
       99    0.019    0.000    0.019    0.000 app.py:85(<listcomp>)
   101027    0.010    0.000    0.010    0.000 {method 'append' of 'list' objects}
       99    0.006    0.000    0.006    0.000 {method 'dot' of 'numpy.ndarray' objects}
      198    0.002    0.000    0.002    0.000 {method 'reduce' of 'numpy.ufunc' objects}
      396    0.000    0.000    0.000    0.000 {method 'reshape' of 'numpy.ndarray' objects}
       99    0.000    0.000    0.000    0.000 {perspectiveTransform}
       99    0.000    0.000    0.000    0.000 {built-in method numpy.array}
       99    0.000    0.000    0.002    0.000 {method 'min' of 'numpy.ndarray' objects}
       99    0.000    0.000    0.001    0.000 {method 'max' of 'numpy.ndarray' objects}
       99    0.000    0.000    0.000    0.000 {built-in method builtins.len}
      198    0.000    0.000    0.000    0.000 {method 'ravel' of 'numpy.ndarray' objects}
       99    0.000    0.000    0.001    0.000 _methods.py:43(_amin)
       99    0.000    0.000    0.001    0.000 _methods.py:39(_amax)
       99    0.000    0.000    0.000    0.000 multiarray.py:153(concatenate)
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}


Temps d'exécution sans CUDA: 27.33 secondes
```

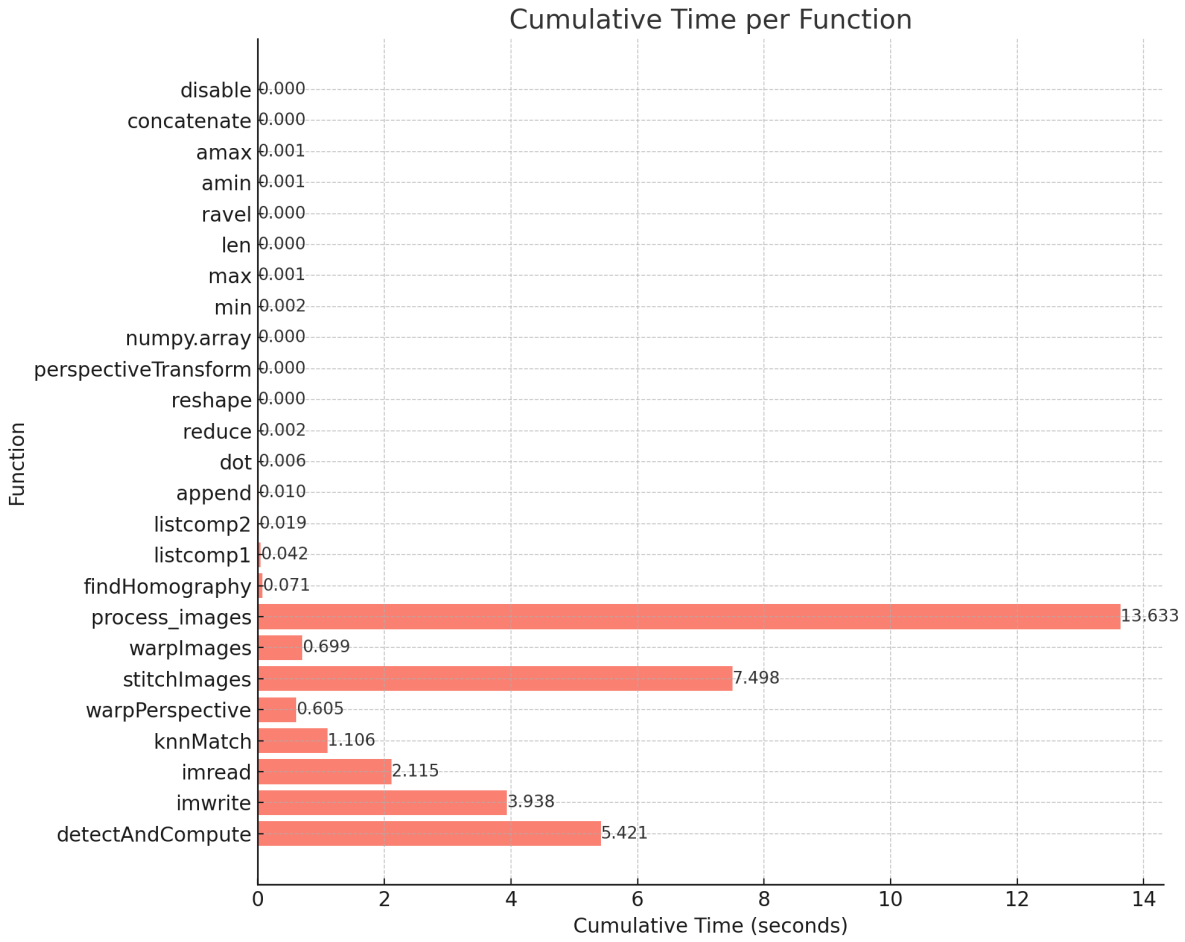
Le résultat est le profil de performance du script Python. Il donne des informations détaillées sur le temps d'exécution de chaque fonction dans le script.

- `ncalls` : le nombre d'appels à la fonction.
- `tottime` : le temps total passé dans la fonction donnée (sans compter le temps passé dans les appels à d'autres fonctions).
- `percall` : le temps moyen passé dans la fonction par appel (tottime/ncalls).
- `cumtime` : le temps cumulé passé dans la fonction donnée, y compris le temps passé dans tous les appels de fonctions à partir de cette fonction.
- `percall` : le temps moyen passé dans la fonction par appel (cumtime/ncalls).
- `filename:lineno(function)` : le nom du fichier, le numéro de ligne et le nom de la fonction.

Dans ce cas, la fonction `detectAndCompute` de l'objet `cv2.Feature2D` est celle qui prend le plus de temps (`tottime`), suivie
par `imwrite`, `imread`, `knnMatch` de `cv2.DescriptorMatcher` et `warpPerspective`.




Total Time per Function
Function
disable
concatenate
amax
amin
ravel
len
max
min
numpy.array
perspectiveTransform
reshape
reduce
dot
append
listcomp2
listcomp1
findHomography
process_images
warpImages
stitchImages
warpPerspective
knnMatch
imread
imwrite
detectAndCompute
0.000
0.000
0.000
0.000
0.000
0.000
0.000
0.000
0.000
0.000
0.000
0.002
0.006
0.010
0.019
0.042
0.071
0.082
0.084
0.130
0.605
1.106
2.115
3.938
5.421
0
1
2
3
4
5
Total Time (seconds)



==========

CPU optimisé imread et imwrite

==========

OUTPUT

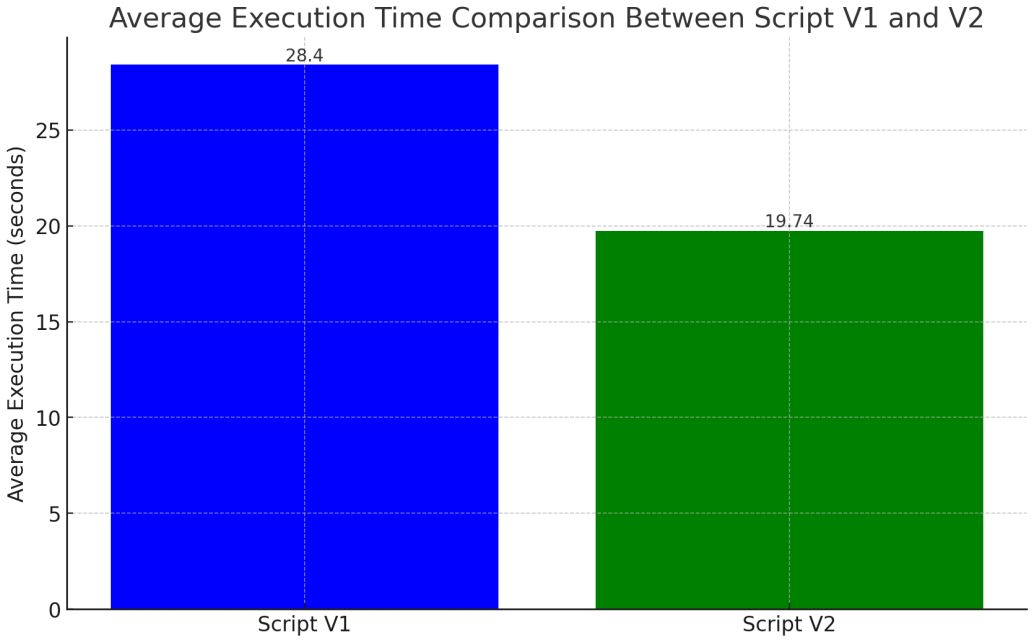
```
(env_gpu) PS C:\Users\ibrah\Desktop\projets\exam_Ti_gpu> python .\ap
         103793 function calls in 7.505 seconds
```

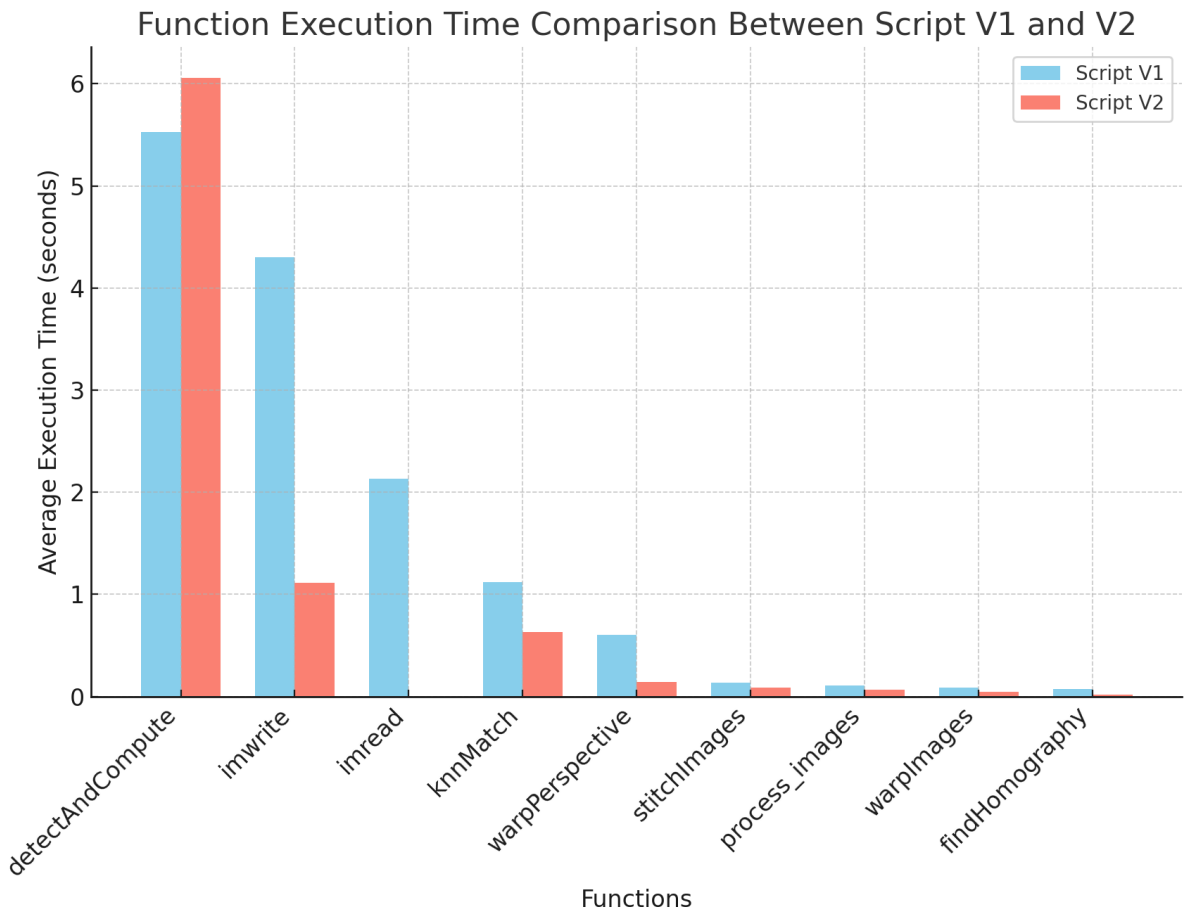
```
   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(functi
      198    5.379    0.027    5.379    0.027 {method 'detectAndComp
       99    1.058    0.011    1.058    0.011 {method 'knnMatch' of
       99    0.588    0.006    0.588    0.006 {warpPerspective}
       99    0.127    0.001    7.388    0.075 app.py:69(stitchImages
       99    0.083    0.001    0.681    0.007 app.py:42(warpImages)
       99    0.072    0.001    0.072    0.001 {findHomography}
        1    0.062    0.062    7.505    7.505 app.py:95(process_imag
        1    0.055    0.055    0.055    0.055 {imwrite}
       99    0.041    0.000    0.041    0.000 app.py:82(<listcomp>)
       99    0.019    0.000    0.019    0.000 app.py:83(<listcomp>)
   101215    0.010    0.000    0.010    0.000 {method 'append' of 'l
       99    0.006    0.000    0.006    0.000 {method 'dot' of 'nump
      198    0.002    0.000    0.002    0.000 {method 'reduce' of 'n
      396    0.000    0.000    0.000    0.000 {method 'reshape' of '
       99    0.000    0.000    0.000    0.000 {perspectiveTransform}
       99    0.000    0.000    0.000    0.000 {built-in method numpy
       99    0.000    0.000    0.002    0.000 {method 'min' of 'nump
       99    0.000    0.000    0.001    0.000 {method 'max' of 'nump
      100    0.000    0.000    0.000    0.000 {built-in method built
      198    0.000    0.000    0.000    0.000 {method 'ravel' of 'nu
       99    0.000    0.000    0.002    0.000 _methods.py:43(_amin)
       99    0.000    0.000    0.001    0.000 _methods.py:39(_amax)
       99    0.000    0.000    0.000    0.000 multiarray.py:153(conc
        1    0.000    0.000    0.000    0.000 {method 'disable' of '


Temps d'exécution sans CUDA: 18.13 secondes
Sun Jan  7 04:04:59 2024
+-----------------------------------------------------------------------
| NVIDIA-SMI 546.01                 Driver Version: 546.01       CUD
|-----------------------------------------+----------------------+--
| GPU  Name                     TCC/WDDM  | Bus-Id        Disp.A | V
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | G
|                                         |                      |
|=========================================+======================+==
|   0  NVIDIA GeForce RTX 2070 ...  WDDM  | 00000000:01:00.0  On |
| N/A   75C    P8             17W /  83W |    646MiB /  8192MiB |
|                                         |                      |
```

```
+-----------------------------------------+----------------------+--

+-----------------------------------------------------------------------
| Processes:
|  GPU   GI   CI        PID   Type   Process name
|        ID   ID
|=======================================================================
|    0   N/A  N/A      1280    C+G   ...t.LockApp_cw5n1h2txyewy\Lock
|    0   N/A  N/A      5064    C+G   ... Access Service\ePowerButton
|    0   N/A  N/A     10068    C+G   C:\Windows\explorer.exe
|    0   N/A  N/A     10540    C+G   ...CBS_cw5n1h2txyewy\TextInputH
|    0   N/A  N/A     13540    C+G   ...nt.CBS_cw5n1h2txyewy\SearchH
|    0   N/A  N/A     13648    C+G   ...2txyewy\StartMenuExperienceH
|    0   N/A  N/A     18732    C+G   ...B\system_tray\lghub_system_t
|    0   N/A  N/A     19076    C+G   C:\Program Files\NordVPN\NordVP
|    0   N/A  N/A     19440    C+G   ...inaries\Win64\EpicGamesLaunc
|    0   N/A  N/A     23812    C+G   ...5n1h2txyewy\ShellExperienceH
|    0   N/A  N/A     41752    C+G   ...crosoft\Edge\Application\mse
|    0   N/A  N/A     60704    C+G   ...ekyb3d8bbwe\PhoneExperienceH
|    0   N/A  N/A     78956    C+G   ...GeForce Experience\NVIDIA Sh
|    0   N/A  N/A     80756    C+G   ...wekyb3d8bbwe\XboxGameBarWidg
|    0   N/A  N/A     85456      C   ...h\anaconda3\envs\env_gpu\pyt
+-----------------------------------------------------------------------

(env_gpu) PS C:\Users\ibrah\Desktop\projets\exam_Ti_gpu> python .\ap
         103793 function calls in 8.069 seconds

   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(functi
      198    5.865    0.030    5.865    0.030 {method 'detectAndComp
       99    1.103    0.011    1.103    0.011 {method 'knnMatch' of
       99    0.611    0.006    0.611    0.006 {warpPerspective}
       99    0.131    0.001    7.954    0.080 app.py:69(stitchImages
       99    0.090    0.001    0.710    0.007 app.py:42(warpImages)
       99    0.072    0.001    0.072    0.001 {findHomography}
        1    0.064    0.064    8.069    8.069 app.py:95(process_imag
        1    0.050    0.050    0.050    0.050 {imwrite}
       99    0.042    0.000    0.042    0.000 app.py:82(<listcomp>)
       99    0.020    0.000    0.020    0.000 app.py:83(<listcomp>)
   101215    0.010    0.000    0.010    0.000 {method 'append' of 'l
       99    0.006    0.000    0.006    0.000 {method 'dot' of 'nump
```

```
      198    0.002    0.000    0.002    0.000 {method 'reduce' of 'n
      396    0.000    0.000    0.000    0.000 {method 'reshape' of '
       99    0.000    0.000    0.000    0.000 {perspectiveTransform}
       99    0.000    0.000    0.000    0.000 {built-in method numpy
       99    0.000    0.000    0.002    0.000 {method 'min' of 'nump
       99    0.000    0.000    0.001    0.000 {method 'max' of 'nump
      100    0.000    0.000    0.000    0.000 {built-in method built
      198    0.000    0.000    0.000    0.000 {method 'ravel' of 'nu
       99    0.000    0.000    0.002    0.000 _methods.py:43(_amin)
       99    0.000    0.000    0.001    0.000 _methods.py:39(_amax)
       99    0.000    0.000    0.000    0.000 multiarray.py:153(conc
        1    0.000    0.000    0.000    0.000 {method 'disable' of '


Temps d'exécution sans CUDA: 18.68 secondes
Sun Jan  7 04:05:25 2024
+-----------------------------------------------------------------------
| NVIDIA-SMI 546.01                 Driver Version: 546.01       CUD
|-----------------------------------------+----------------------+--
| GPU  Name                     TCC/WDDM  | Bus-Id        Disp.A | V
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | G
|                                         |                      |
|=========================================+======================+==
|   0  NVIDIA GeForce RTX 2070 ...  WDDM  | 00000000:01:00.0  On |
| N/A   76C    P8              9W /  81W |    646MiB /  8192MiB |
|                                         |                      |
+-----------------------------------------+----------------------+--

+-----------------------------------------------------------------------
| Processes:
|  GPU   GI   CI        PID   Type   Process name
|        ID   ID
|=======================================================================
|    0   N/A  N/A      1280    C+G   ...t.LockApp_cw5n1h2txyewy\Lock
|    0   N/A  N/A      5064    C+G   ... Access Service\ePowerButton
|    0   N/A  N/A     10068    C+G   C:\Windows\explorer.exe
|    0   N/A  N/A     10540    C+G   ...CBS_cw5n1h2txyewy\TextInputH
|    0   N/A  N/A     13540    C+G   ...nt.CBS_cw5n1h2txyewy\SearchH
|    0   N/A  N/A     13648    C+G   ...2txyewy\StartMenuExperienceH
|    0   N/A  N/A     18732    C+G   ...B\system_tray\lghub_system_t
|    0   N/A  N/A     19076    C+G   C:\Program Files\NordVPN\NordVP
|    0   N/A  N/A     19440    C+G   ...inaries\Win64\EpicGamesLaunc
```

```
|    0   N/A  N/A     23812    C+G   ...5n1h2txyewy\ShellExperienceH
|    0   N/A  N/A     41752    C+G   ...crosoft\Edge\Application\mse
|    0   N/A  N/A     60704    C+G   ...ekyb3d8bbwe\PhoneExperienceH
|    0   N/A  N/A     78956    C+G   ...GeForce Experience\NVIDIA Sh
|    0   N/A  N/A     80756    C+G   ...wekyb3d8bbwe\XboxGameBarWidg
|    0   N/A  N/A     82764      C   ...h\anaconda3\envs\env_gpu\pyt
+-----------------------------------------------------------------------------

(env_gpu) PS C:\Users\ibrah\Desktop\projets\exam_Ti_gpu>
```





Algorithme de detection ORB

`detectAndCompute` —> Keypoint et Descripteurs

Pour detecter les kp et Dscp

Kp = mini region depixel unique / intéressant —→ en `print(keypoints[0])` on a ca : `<KeyPoint 0x7f8e6c8>`

- **kp**: Ce sont des points spécifiques dans une image qui sont uniques et peuvent être utilisés pour comparer différentes images. Par exemple, les coins, les bords et les taches sont souvent utilisés comme points d'intérêt car ils sont uniques et peuvent être facilement identifiés dans différentes images.

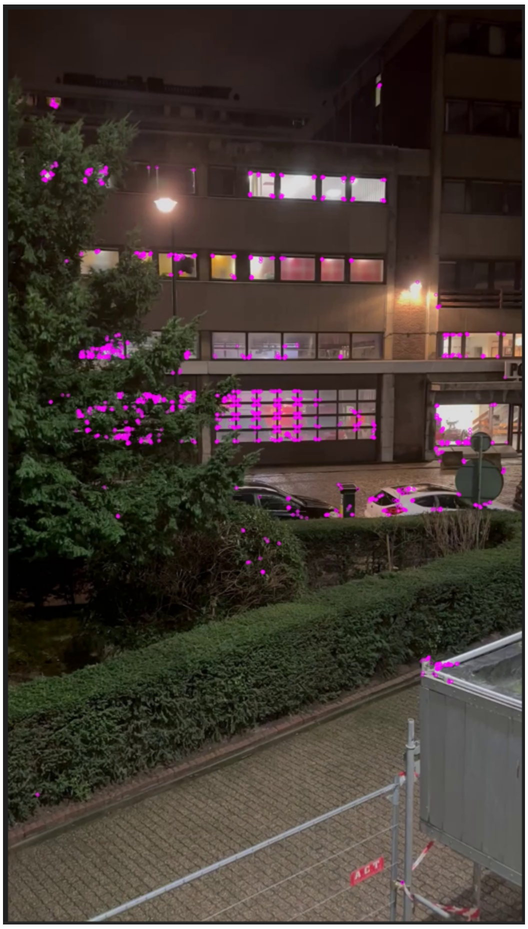
  il continnent les info comme la position du point d'intérêt (x, y), la taille de la région d'intérêt

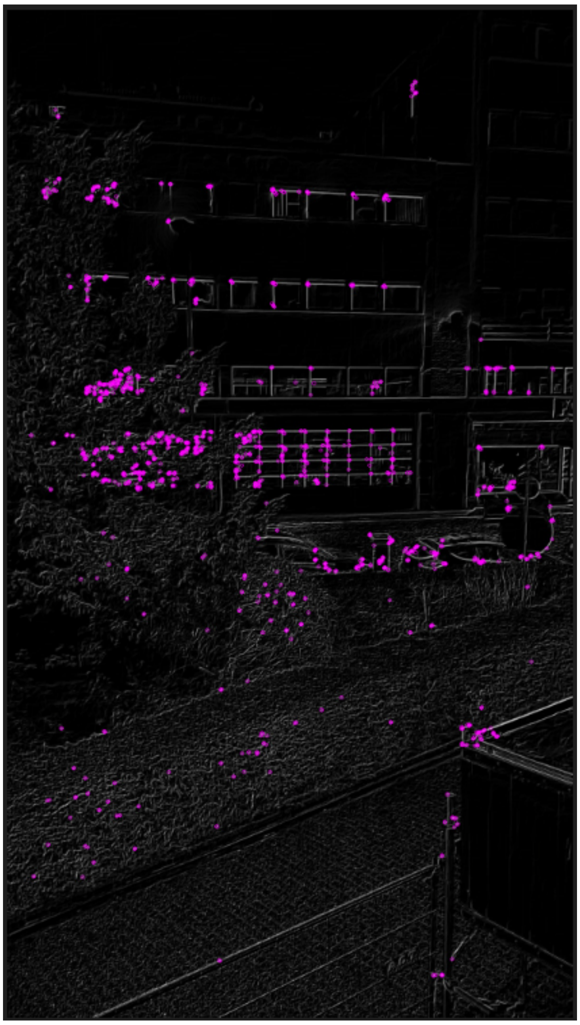
- **Dscp** : est représentation numérique d'un kp et donc sert a comparer les points entre 2 img

`print(descriptors[0])` [49 241 236 253 255 159 254 223 63 254 255 254 255 127 191 251 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255]

Pour `orb = cv2.ORB_create(nfeatures=2000)`

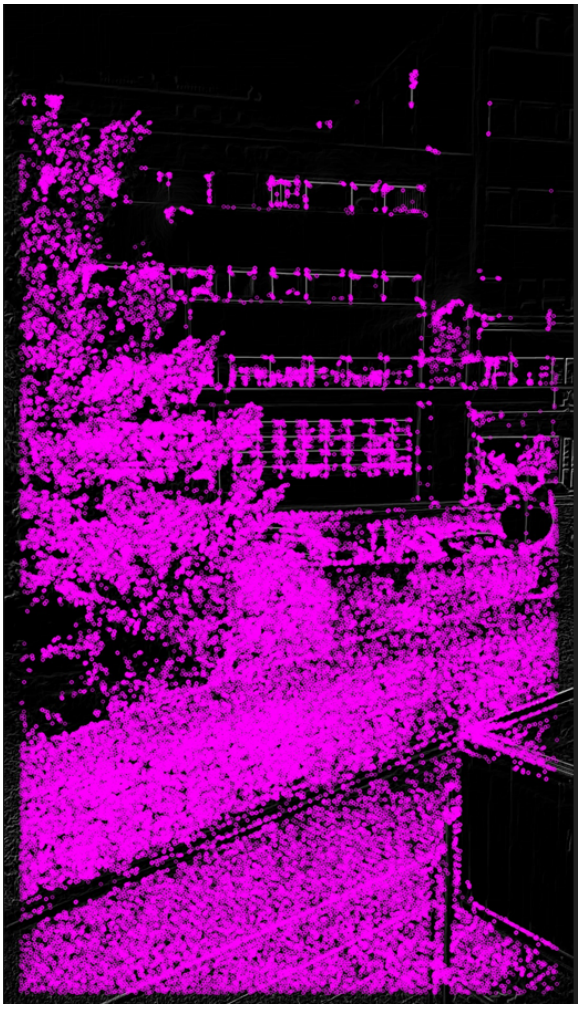
Temps d'exécution : 0.35 secondes en moyenne

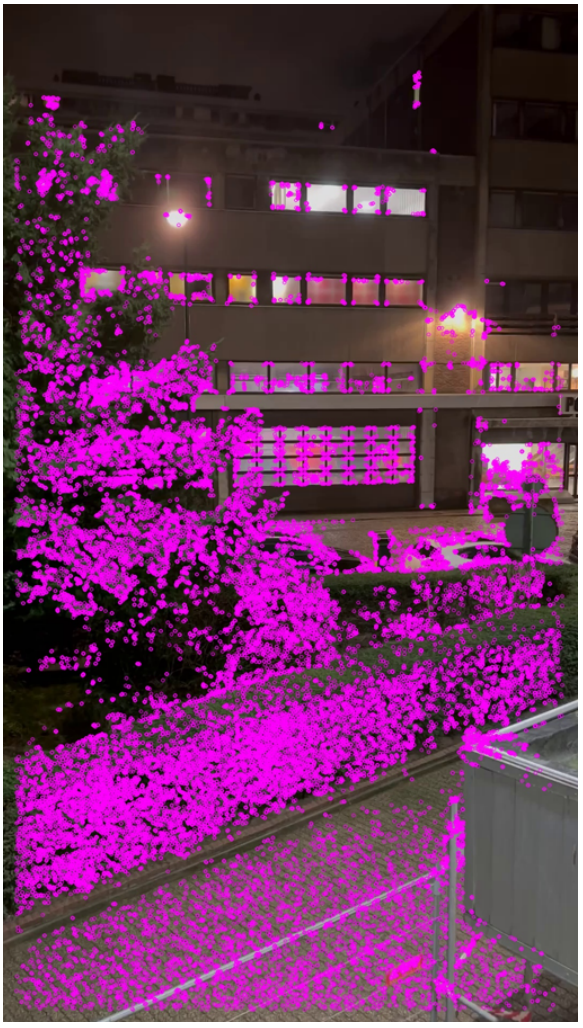




`A 200000`

Temps d'exécution : 0.56 secondes





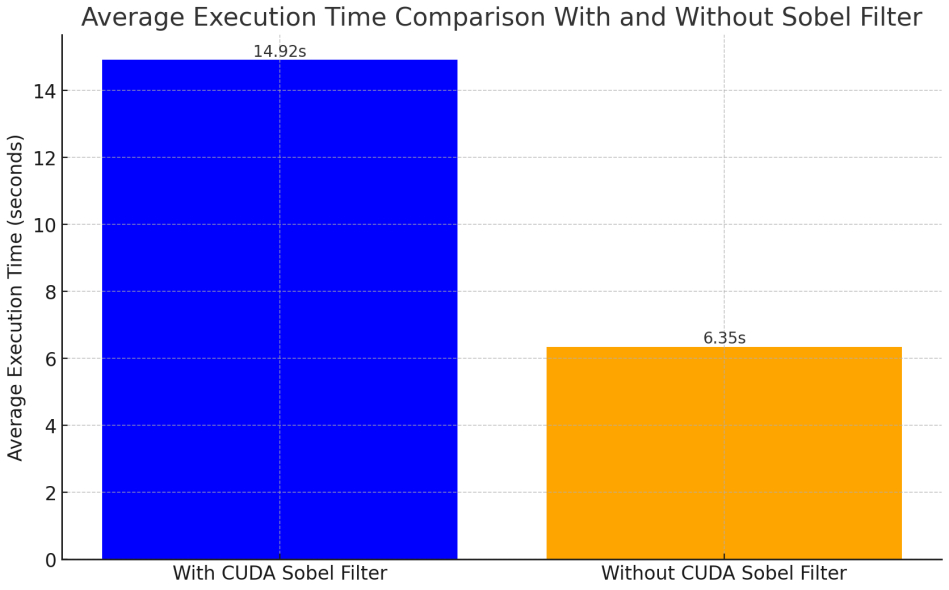
UTILISATION DE *CUDA* POUR ENCORE OPTIMISER LE SCRIPT GRACE A LA PUISSANCE DU *GPU*

Filtre de Sobel pour aider les algo de detection ici ORB

Il aide à mieux détecter les bordures d’images ce qui peut rendre l'alignement des images pour la construction du panorama plus précis

Avec filtre sobel

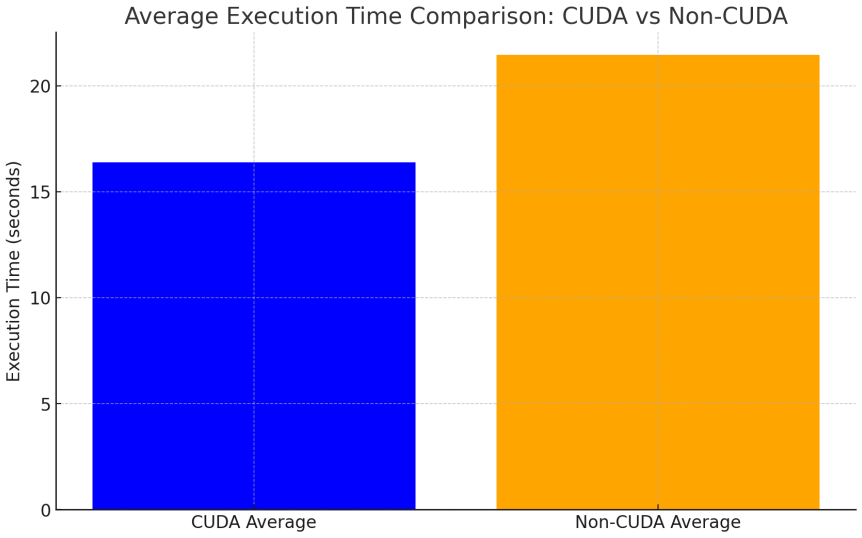
Sans filtre sobel

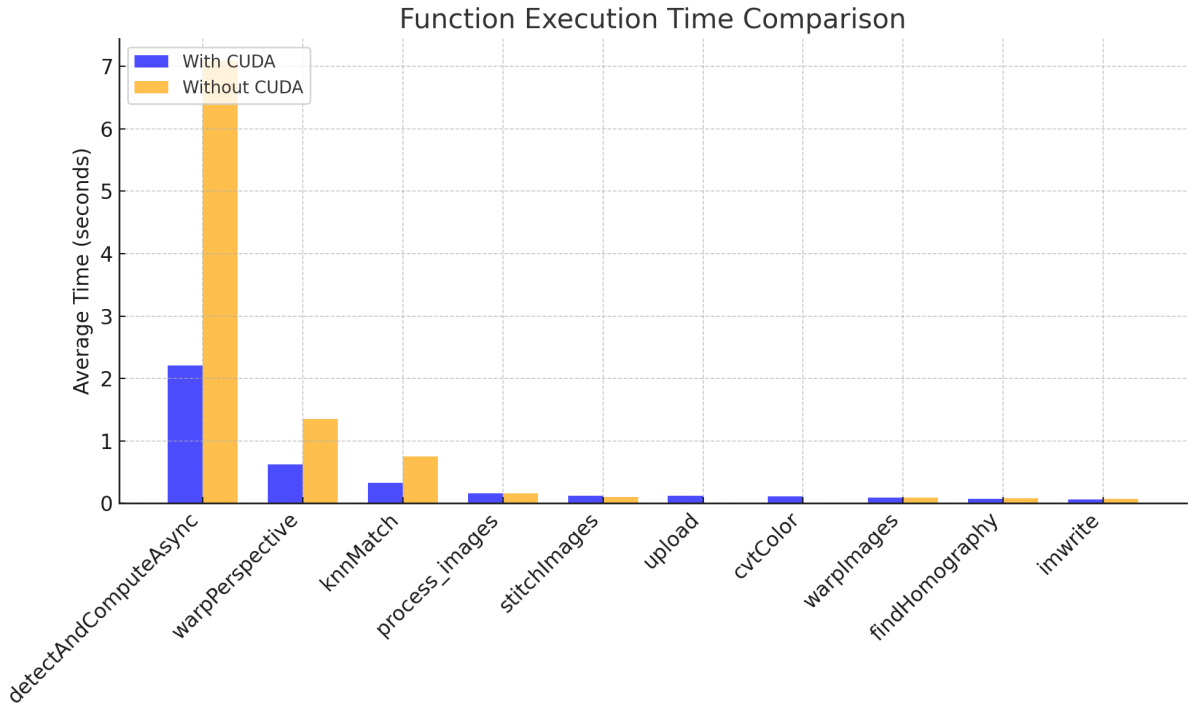


`detectAndCompute` faire tout cet ensemle en gpu et renvoyer les resultat au cpu

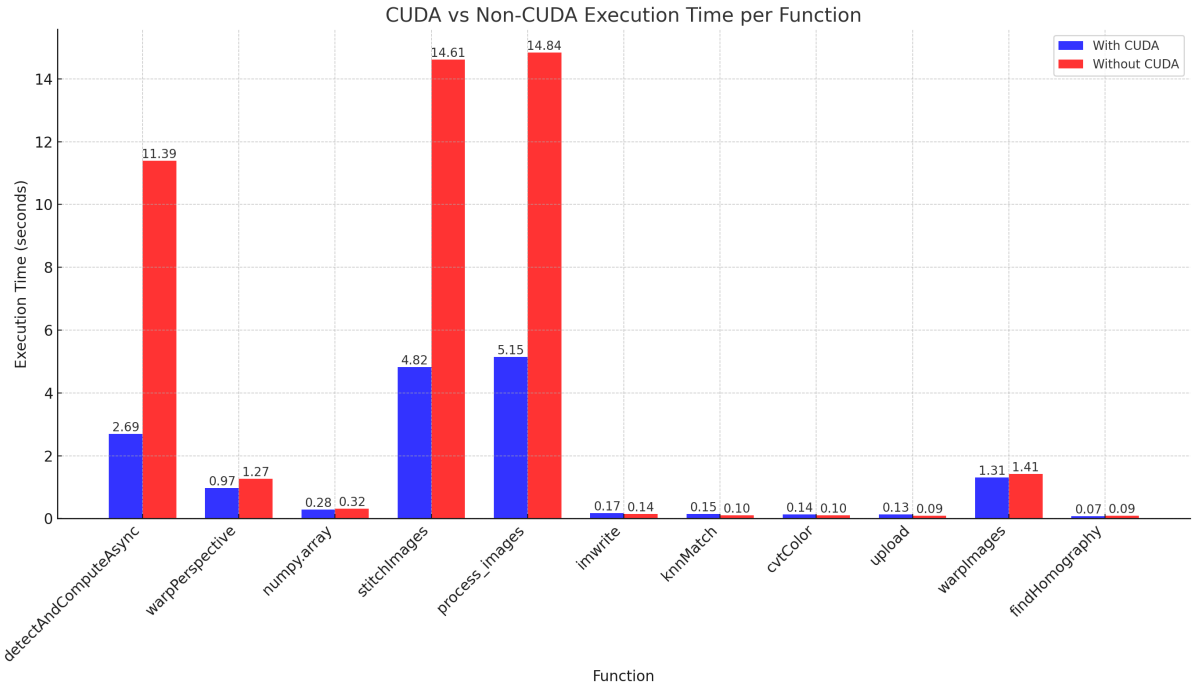
`matches = bf.knnMatch(descriptors1, descriptors2, k=2)` envoie directement le resultat au cpu

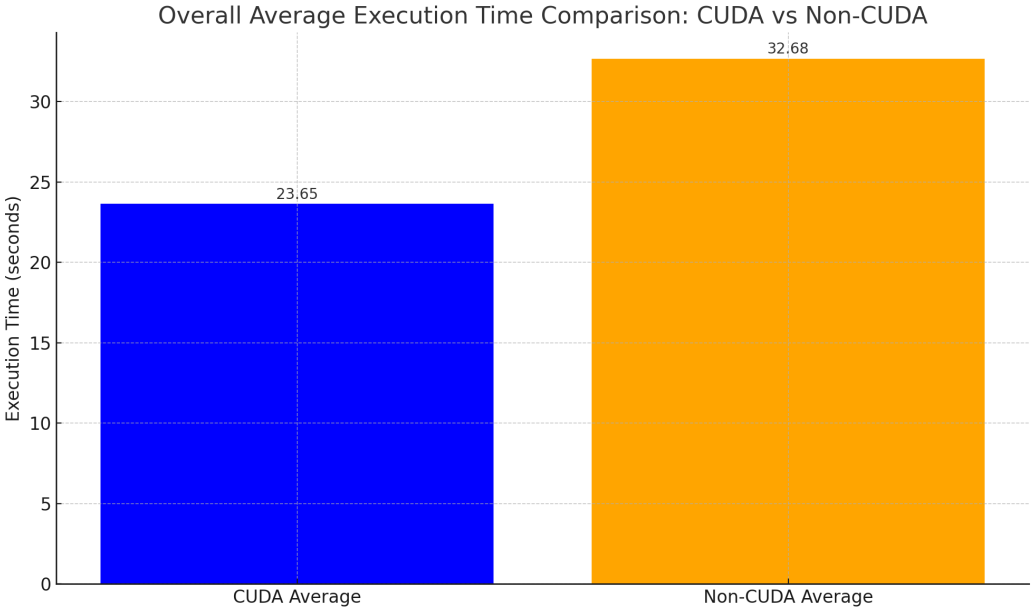
==========Video2==========



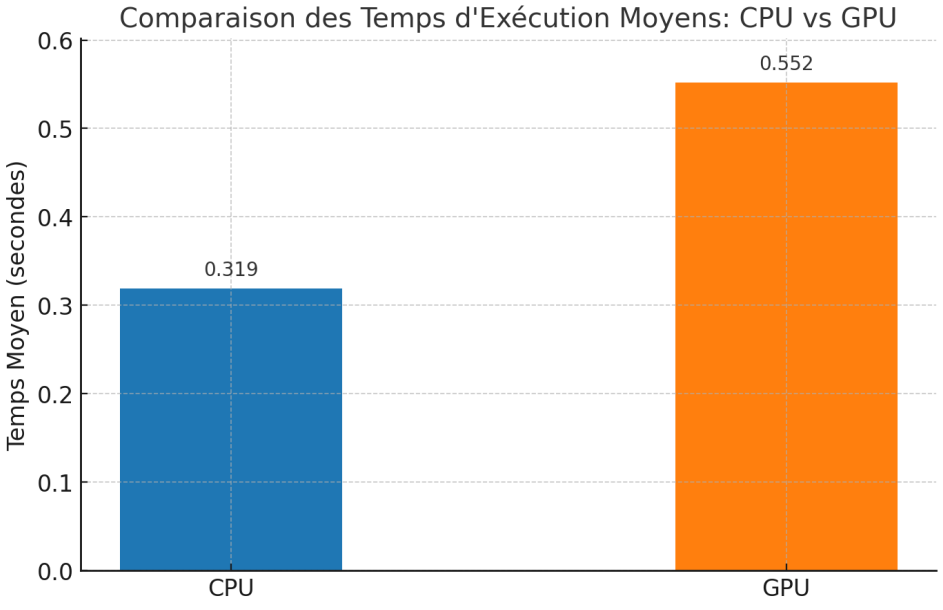

Function Execution Time Comparison
With CUDA
Without CUDA
Average Time (seconds)
0
1
2
3
4
5
6
7
detectAndComputeAsync
warpPerspective
knnMatch
process_images
stitchImages
upload
cvtColor
warpImages
findHomography
imwrite

========== Video1 ==========


CUDA vs Non-CUDA Execution Time per Function
With CUDA
Without CUDA
Execution Time (seconds)
0
2
4
6
8
10
12
14
2.69
11.39
0.97
1.27
0.28
0.32
4.82
14.61
5.15
14.84
0.17
0.14
0.15
0.10
0.14
0.10
0.13
0.09
1.31
1.41
0.07
0.09
detectAndComputeAsync
warpPerspective
numpy.array
stitchImages
process_images
imwrite
knnMatch
cvtColor
upload
warpImages
findHomography
Function



CuPy vs Numpy



==========

PANORAMA

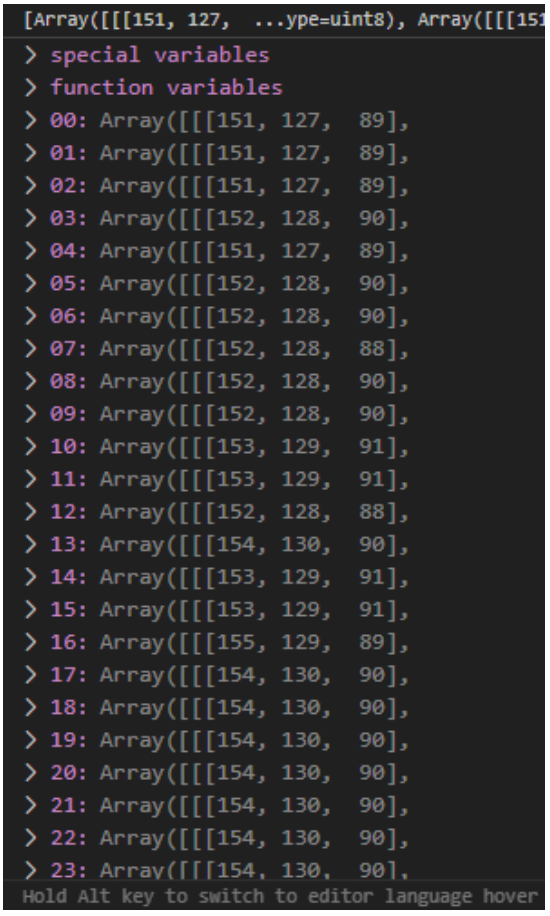
==========

Image

Une `image` c’est grosse matrice de pixels.

```
[138 140 158]
[136 138 156]
[127 129 147]
```

Chaque ligne de la matrice représente un pixel donc numpy va etre important pour la suite



`cv2.imread()` Va lire l'image et la convertit en une matrice

Algo de detection de point clé —> ORB

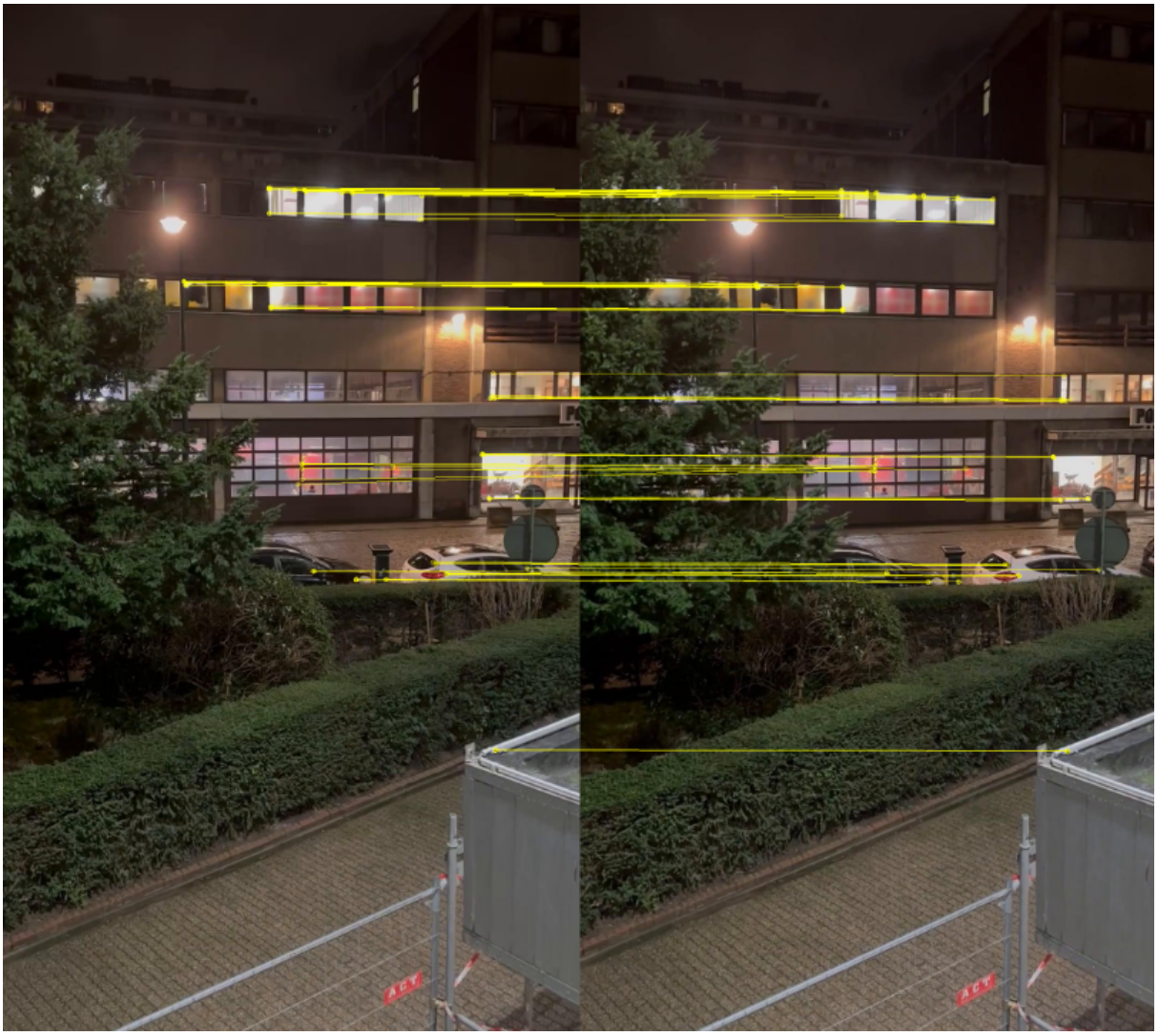
OpenCV: ORB (Oriented FAST and Rotated BRIEF)

In this chapter,

https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html

Trouver des régions ou les img match, et appliquer une transformation de perspective pourqu'elle soit positionner sur le meme plan ou la premiere sera la cadre de ref et la 2em l'image déformé, de maniere a aligné les caratéristique des 2 images

Récupérer les données qui match



Superposition

```
min_coords = list_of_points.min(axis=0).ravel() - 0.5 #j'aplati le tableau et je soustrait 0.5
```