Computer Vision: COMP 6341
Student Id: 40067635
Assignment 2 Result Report:

There are three main parts of the assignment:

1. Feature Detector
2. Feature descriptor
3. Feature matching

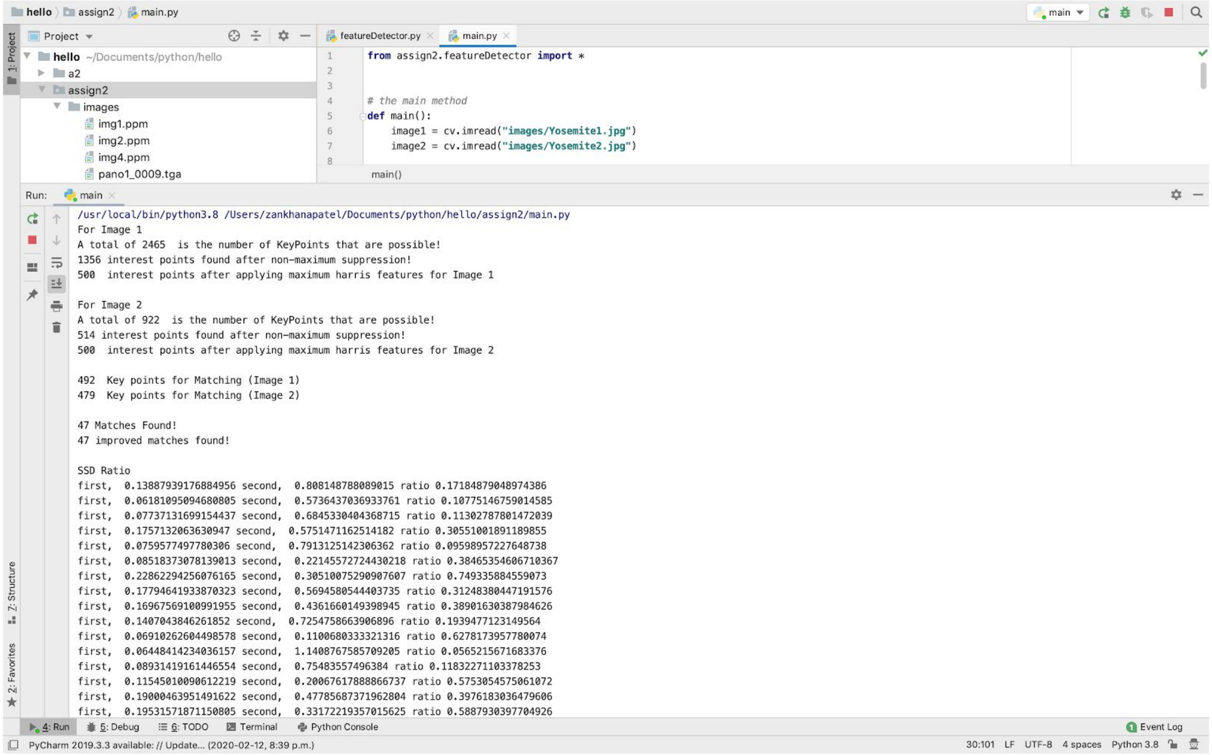
I have implemented all 3 parts in the python language. I am reporting below some significant results of the assignment derived from my code. The bold letters are results directly from my console.

I have used 2 images of **Yosemite1.jpg and Yosemite2.jpg** for my implementation.

1. Feature Detector

   I have used the Harris corner detector to detect corners and main interest points from both images. I have also found local maxima and adaptive maximum suppression.

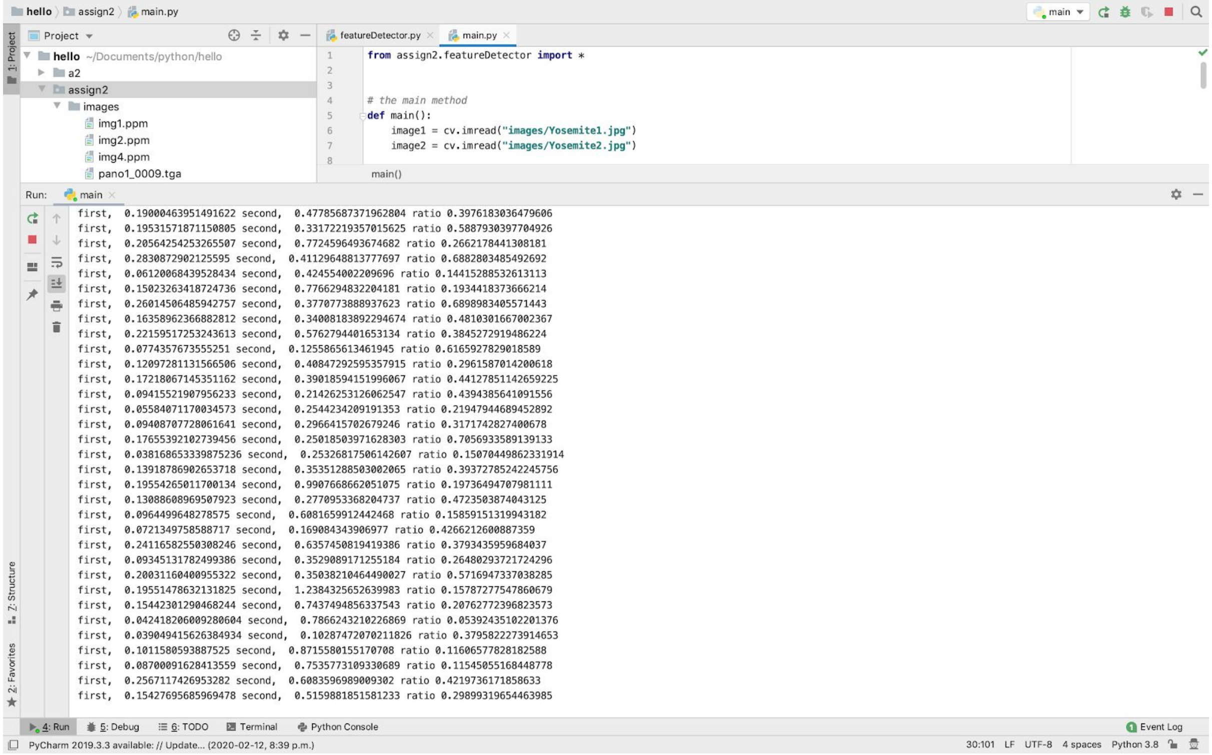
   The result from the console:



```python
from assign2.featureDetector import *


# the main method
def main():
    image1 = cv.imread("images/Yosemite1.jpg")
    image2 = cv.imread("images/Yosemite2.jpg")

```

```
/usr/local/bin/python3.8 /Users/zankhanapatel/Documents/python/hello/assign2/main.py
For Image 1
A total of 2465  is the number of KeyPoints that are possible!
1356 interest points found after non-maximum suppression!
500  interest points after applying maximum harris features for Image 1

For Image 2
A total of 922  is the number of KeyPoints that are possible!
514 interest points found after non-maximum suppression!
500  interest points after applying maximum harris features for Image 2

492  Key points for Matching (Image 1)
479  Key points for Matching (Image 2)

47 Matches Found!
47 improved matches found!

SSD Ratio
first,  0.13887939176884956 second,  0.808148788089015 ratio 0.17184879048974386
first,  0.06181095094680805 second,  0.5736437036933761 ratio 0.10775146759014585
first,  0.07737131699154437 second,  0.6845330404368715 ratio 0.11302787801472039
first,  0.1757132063630947 second,  0.575147116251418 2 ratio 0.30551001891189855
first,  0.0759577497780306 second,  0.7913125142306362 ratio 0.09598957227648738
first,  0.08518373078139013 second,  0.22145572724430218 ratio 0.38465354606710367
first,  0.22862294256076165 second,  0.30510075290907607 ratio 0.749335884559073
first,  0.17794641933870323 second,  0.5694580544403735 ratio 0.31248380447191576
first,  0.16967569100991955 second,  0.4361660149398945 ratio 0.38901630387984626
first,  0.1407043846261852 second,  0.7254758663906896 ratio 0.193947712314956 4
first,  0.06910262604498578 second,  0.1100680333321316 ratio 0.6278173957780074
first,  0.06448414234036157 second,  1.1408767585709205 ratio 0.0565215671683376
first,  0.08931419161446554 second,  0.75483557496384 ratio 0.11832271103378253
first,  0.11545010090612219 second,  0.20067617888866737 ratio 0.5753054575061072
first,  0.19000463951491622 second,  0.47785687371962804 ratio 0.3976183036479606
first,  0.19531571871150805 second,  0.33172219357015625 ratio 0.5887930397704926
```

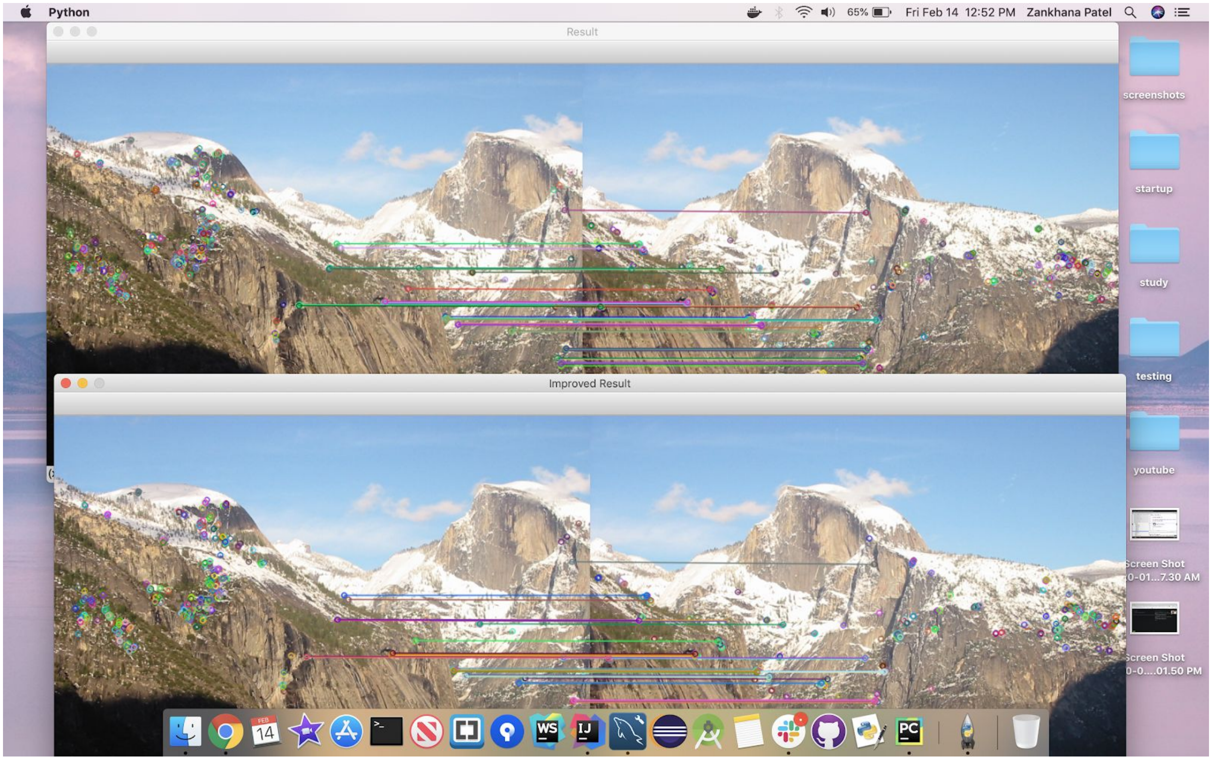
Screenshot 1



```python
from assign2.featureDetector import *


# the main method
def main():
    image1 = cv.imread("images/Yosemite1.jpg")
    image2 = cv.imread("images/Yosemite2.jpg")

```

```
first,  0.19000463951491622 second,  0.47785687371962804 ratio 0.3976183036479606
first,  0.19531571871150805 second,  0.33172219357015625 ratio 0.5887930397704926
first,  0.20564254253265507 second,  0.7724596493674682 ratio 0.2662178441308181
first,  0.2830872902125595 second,  0.41129648813777697 ratio 0.6882803485492692
first,  0.06120068439528434 second,  0.424554002209696 ratio 0.14415288532613113
first,  0.15023263418724736 second,  0.7766294832204181 ratio 0.1934418373666214
first,  0.26014506485942757 second,  0.3770773888937623 ratio 0.6898983405571443
first,  0.16358962366882812 second,  0.34008183892294674 ratio 0.4810301667002367
first,  0.22159517253243613 second,  0.5762794401653134 ratio 0.3845272919486224
first,  0.0774357673555251 second,  0.1255865613461945 ratio 0.6165927829018589
first,  0.12097281131566506 second,  0.40847292595357915 ratio 0.2961587014200618
first,  0.17218067145351162 second,  0.39018594151996067 ratio 0.44127851142659225
first,  0.09415521907956233 second,  0.21426253126062547 ratio 0.4394385641091556
first,  0.05584071170034573 second,  0.2544234209191353 ratio 0.21947944689452892
first,  0.09408707728061641 second,  0.2966415702679246 ratio 0.3171742827400678
first,  0.17655392102739456 second,  0.25018503971628303 ratio 0.7056933589139133
first,  0.038168653339875236 second,  0.25326817506142607 ratio 0.15070449862331914
first,  0.13918786902653718 second,  0.35351288503002065 ratio 0.39372785242245756
first,  0.19554265011700134 second,  0.9907668662051075 ratio 0.19736494707981111
first,  0.13088608969507923 second,  0.2770953368204737 ratio 0.4723503874043125
first,  0.0964499648278575 second,  0.6081659912442468 ratio 0.15859151319943182
first,  0.0721349758588717 second,  0.169084343906977 ratio 0.4266212600887359
first,  0.24116582550308246 second,  0.6357450819419386 ratio 0.3793435959684037
first,  0.09345131782499386 second,  0.3529089171255184 ratio 0.26480293721724296
first,  0.20031160400955322 second,  0.35038210464490027 ratio 0.5716947337038285
first,  0.19551478632131825 second,  1.2384325652639983 ratio 0.15787277547860679
first,  0.15442301290468244 second,  0.7437494856337543 ratio 0.20762772396823573
first,  0.042418206009280604 second,  0.7866243210226869 ratio 0.05392435102201376
first,  0.039049415626384934 second,  0.10287472070211826 ratio 0.3795822273914653
first,  0.1011580593887525 second,  0.8715580155170708 ratio 0.11606577828182588
first,  0.08700091628413559 second,  0.7535773109330689 ratio 0.11545055168448778
first,  0.2567117426953282 second,  0.6083596989009302 ratio 0.4219736171858633
first,  0.15427695685969478 second,  0.5159881851581233 ratio 0.29899319654463985
```

Screenshot 2



Screenshot 3

2. Feature descriptor
   I have used the SIFT descriptor and rotation invariance took into account. Histograms made for both images to compare with each other. After normalizing the values I got total points from both images for matching.
   The result is shown in screenshot 1 and 4.

3. Feature matching
   Best Feature matched with SSDRatio between interest points of image 1 and image 2. I found SSD (Sum of Squared Differences) for each point for images and then find the best match and second-best match key point.

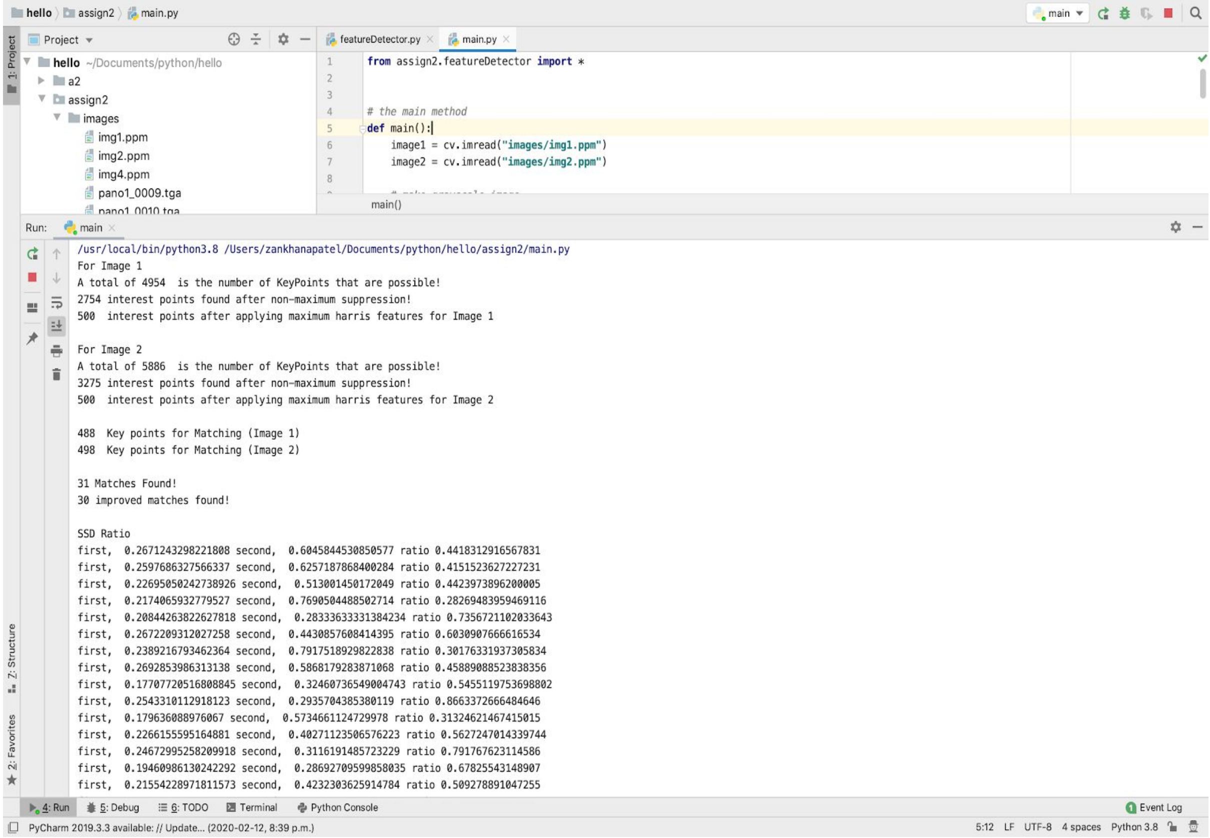
   The result is shown in screenshot 1, 2 and 3.

   I have found improved matches also between 2 images. Which is giving less number of points other than the previous match found technique (not for all images). I threshold the ratio to 0.8 to find only best matches.
   The result is shown in screenshot 4 for the images img1.ppm and img2.ppm.

   Furthermore, I have printed the SSD ratio between the matched point from both images. You can see it on screenshot 1,2 and 4,5.

Point to be noted for the improved match:
I cannot find an improved match according to the ratio test for Yosemite images. However, I can find improved matches in img1.ppm and img2.ppm.

I have reported the result below:

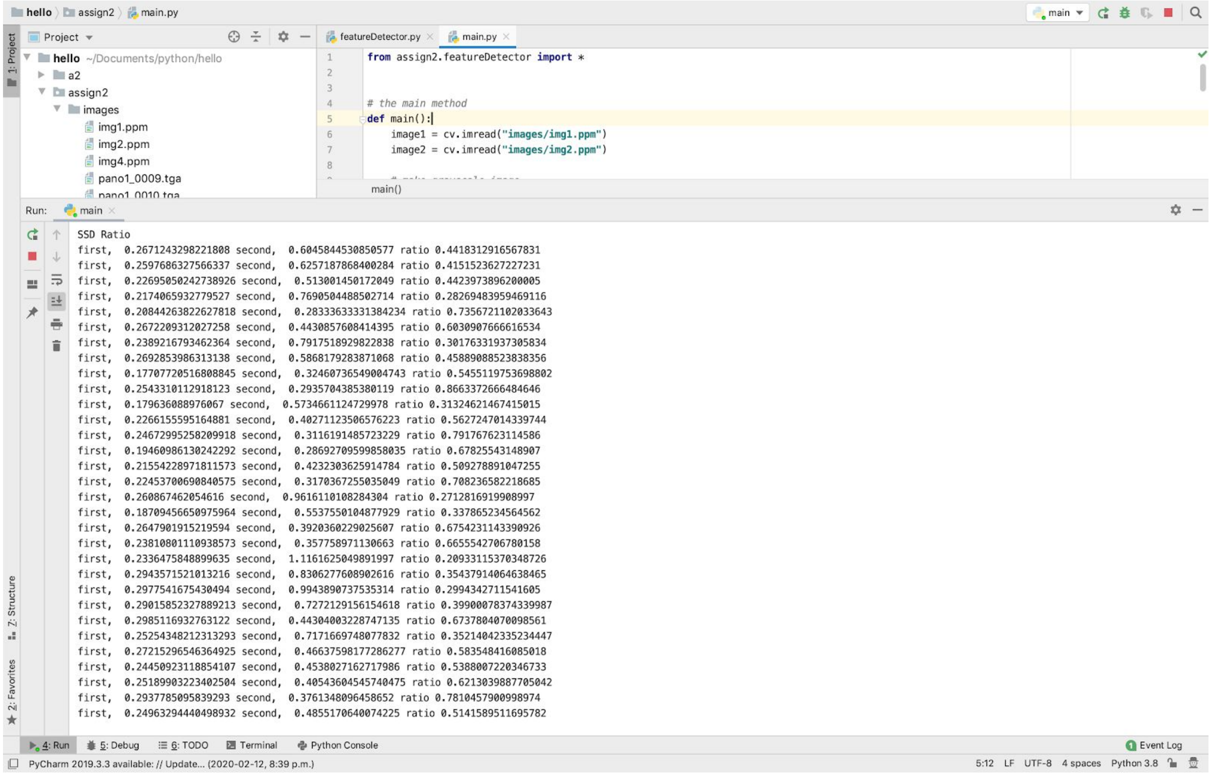


```
from assign2.featureDetector import *


# the main method
def main():
    image1 = cv.imread("images/img1.ppm")
    image2 = cv.imread("images/img2.ppm")
```

```
/usr/local/bin/python3.8 /Users/zankhanapatel/Documents/python/hello/assign2/main.py
For Image 1
A total of 4954  is the number of KeyPoints that are possible!
2754 interest points found after non-maximum suppression!
500  interest points after applying maximum harris features for Image 1

For Image 2
A total of 5886  is the number of KeyPoints that are possible!
3275 interest points found after non-maximum suppression!
500  interest points after applying maximum harris features for Image 2

488  Key points for Matching (Image 1)
498  Key points for Matching (Image 2)

31 Matches Found!
30 improved matches found!

SSD Ratio
first,  0.2671243298221808 second,  0.6045844530850577 ratio 0.4418312916567831
first,  0.2597686327566337 second,  0.6257187868400284 ratio 0.4151523627227231
first,  0.22695050242738926 second,  0.513001450172049 ratio 0.4423973896200005
first,  0.2174065932779527 second,  0.7690504488502714 ratio 0.28269483959469116
first,  0.20844263822627818 second,  0.28333633331384234 ratio 0.7356721102033643
first,  0.2672209312027258 second,  0.4430857608414395 ratio 0.6030907666616534
first,  0.2389216793462364 second,  0.7917518929822838 ratio 0.30176331937305834
first,  0.2692853986313138 second,  0.5868179283871068 ratio 0.45889088523838356
first,  0.17707720516808845 second,  0.32460736549004743 ratio 0.5455119753698802
first,  0.2543310112918123 second,  0.2935704385380119 ratio 0.8663372666484646
first,  0.179636088976067 second,  0.5734661124729978 ratio 0.31324621467415015
first,  0.2266155595164881 second,  0.40271123506576223 ratio 0.5627247014339744
first,  0.24672995258209918 second,  0.3116191485723229 ratio 0.791767623114586
first,  0.19460986130242292 second,  0.28692709599858035 ratio 0.67825543148907
first,  0.21554228971811573 second,  0.4232303625914784 ratio 0.509278891047255
```

PyCharm 2019.3.3 available: // Update... (2020-02-12, 8:39 p.m.)   5:12 LF UTF-8 4 spaces Python 3.8

Screenshot 4



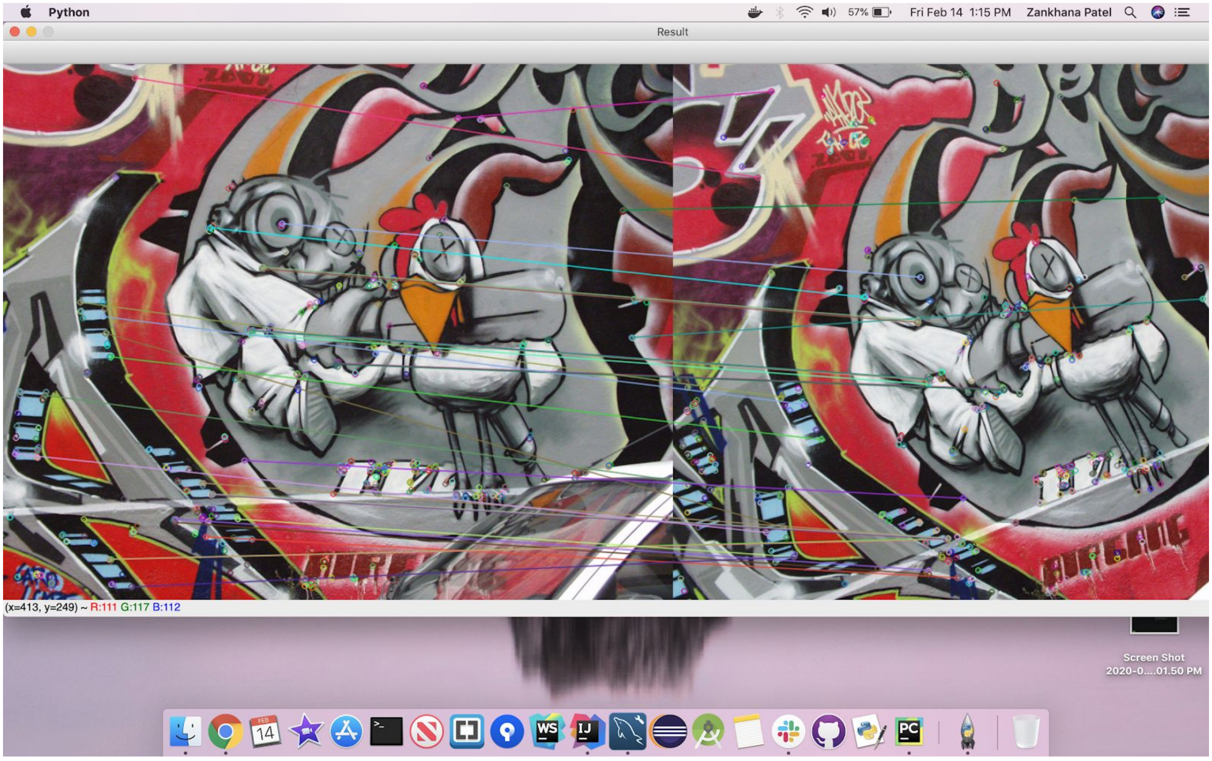
```
from assign2.featureDetector import *


# the main method
def main():
    image1 = cv.imread("images/img1.ppm")
    image2 = cv.imread("images/img2.ppm")
```

```
SSD Ratio
first,  0.2671243298221808 second,  0.6045844530850577 ratio 0.4418312916567831
first,  0.2597686327566337 second,  0.6257187868400284 ratio 0.4151523627227231
first,  0.22695050242738926 second,  0.513001450172049 ratio 0.4423973896200005
first,  0.2174065932779527 second,  0.7690504488502714 ratio 0.28269483959469116
first,  0.20844263822627818 second,  0.28333633331384234 ratio 0.7356721102033643
first,  0.2672209312027258 second,  0.4430857608414395 ratio 0.6030907666616534
first,  0.2389216793462364 second,  0.7917518929822838 ratio 0.30176331937305834
first,  0.2692853986313138 second,  0.5868179283871068 ratio 0.45889088523838356
first,  0.17707720516808845 second,  0.32460736549004743 ratio 0.5455119753698802
first,  0.2543310112918123 second,  0.2935704385380119 ratio 0.8663372666484646
first,  0.179636088976067 second,  0.5734661124729978 ratio 0.31324621467415015
first,  0.2266155595164881 second,  0.40271123506576223 ratio 0.5627247014339744
first,  0.24672995258209918 second,  0.3116191485723229 ratio 0.791767623114586
first,  0.19460986130242292 second,  0.28692709599858035 ratio 0.67825543148907
first,  0.21554228971811573 second,  0.4232303625914784 ratio 0.509278891047255
first,  0.22453700690840575 second,  0.3170367255035049 ratio 0.708236582218685
first,  0.260867462054616 second,  0.9616110108284304 ratio 0.2712816919908997
first,  0.18709456650975964 second,  0.5537550104877929 ratio 0.337865234564562
first,  0.2647901915219594 second,  0.3920360229025607 ratio 0.6754231143390926
first,  0.23810801110938573 second,  0.357758971130663 ratio 0.6655542706780158
first,  0.2336475848899635 second,  1.1161625049891997 ratio 0.20933115370348726
first,  0.2943571521013216 second,  0.8306277608902616 ratio 0.35437914064638465
first,  0.2977541675430494 second,  0.9943890737535314 ratio 0.2994342711541605
first,  0.29015852327889213 second,  0.7272129156154618 ratio 0.39900078374339987
first,  0.2985116932763122 second,  0.44304003228747135 ratio 0.6737804070098561
first,  0.25254348212313293 second,  0.7171669748077832 ratio 0.35214042335234447
first,  0.27215296546364925 second,  0.46637598177286277 ratio 0.583548416085018
first,  0.24450923118854107 second,  0.4538027162717986 ratio 0.5388007220346733
first,  0.25189903223402504 second,  0.40543604545740475 ratio 0.6213039887705042
first,  0.2937785095839293 second,  0.3761348096458652 ratio 0.7810457900998974
first,  0.24963294440498932 second,  0.4855170640074225 ratio 0.5141589511695782
```

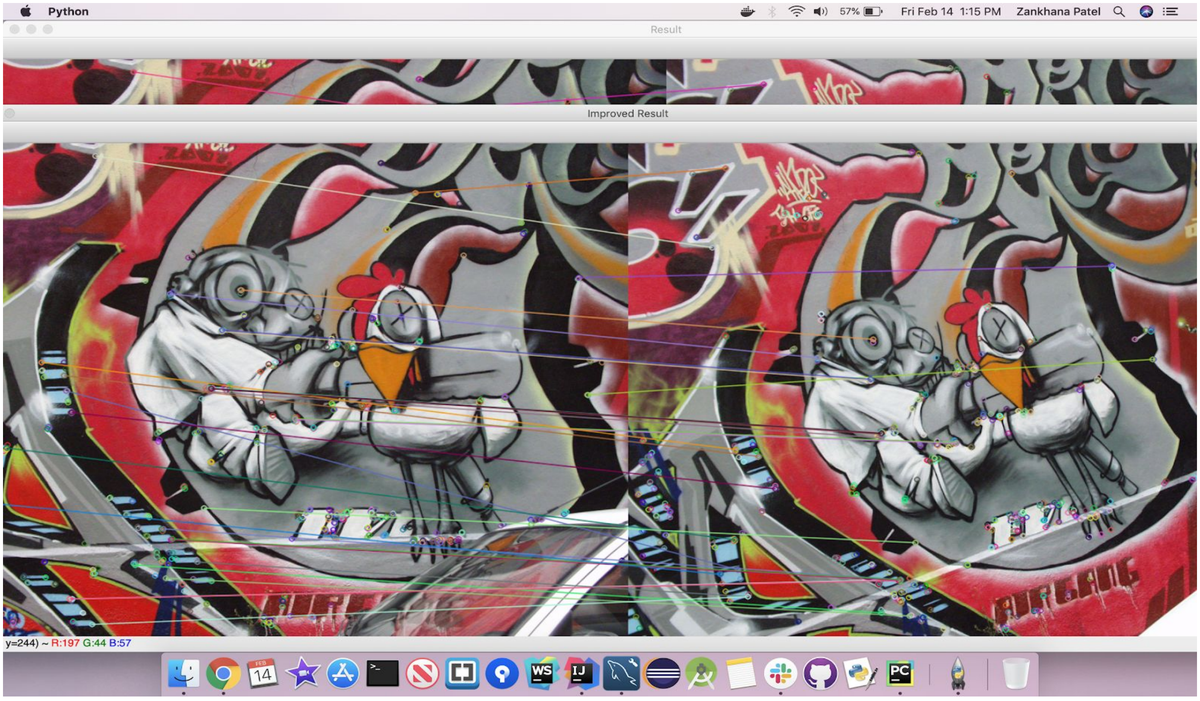
PyCharm 2019.3.3 available: // Update... (2020-02-12, 8:39 p.m.)   5:12 LF UTF-8 4 spaces Python 3.8

Screenshot 5

The matched point image screenshot:


Python
Fri Feb 14 1:15 PM Zankhana Patel
Result
(x=413, y=249) ~ R:111 G:117 B:112
Screen Shot 2020-0....01.50 PM

Screenshot 6



Screenshot 7