CS5542 – Bigdata Analytics and Applications

Lab report – 1

Submitted by:

Saitejaswi Koppuravuri – 13

Team – 5

Objectives:

The objectives of the Lab assignment 1:

- Downloading the dataset according to the idea of the project proposal.
- Performing tokenization and lemmatization on the caption or text data.
- Reporting the image statistics from the caption data extracted.
- Perform feature extraction using SIFT algorithm on the image data extracted.

Technologies:

Pycharm – IDE for executing the python files

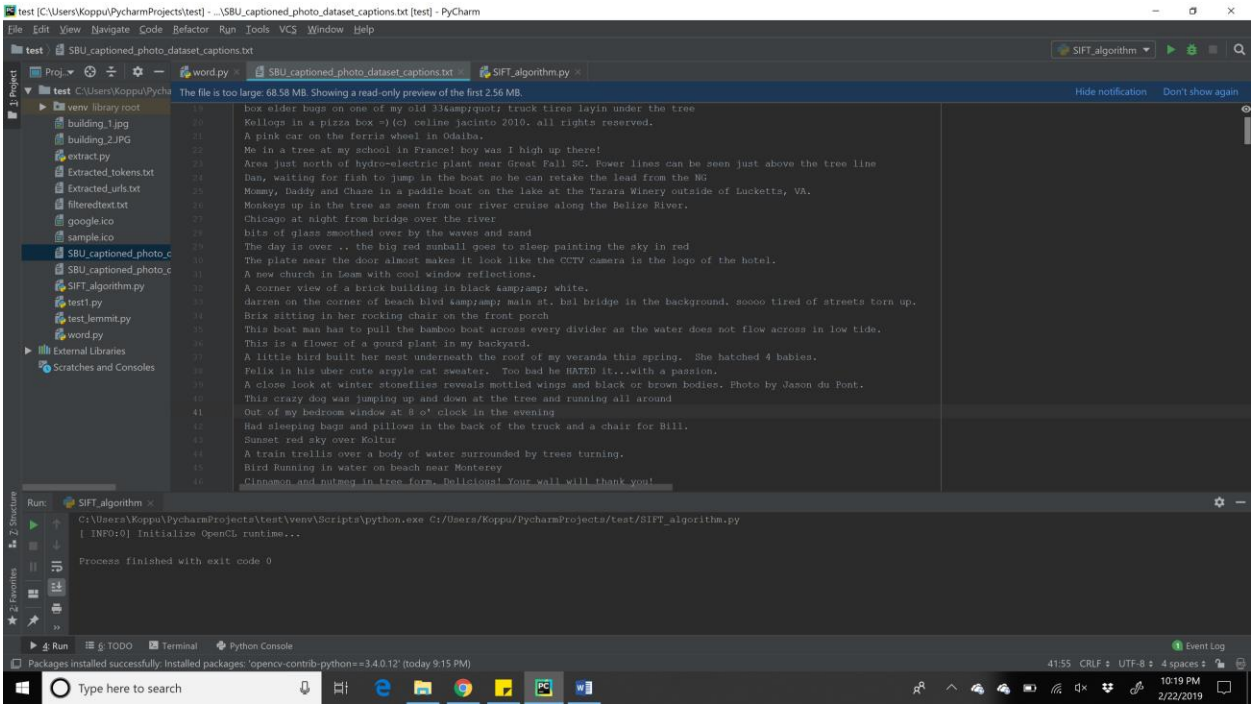
Packages used:

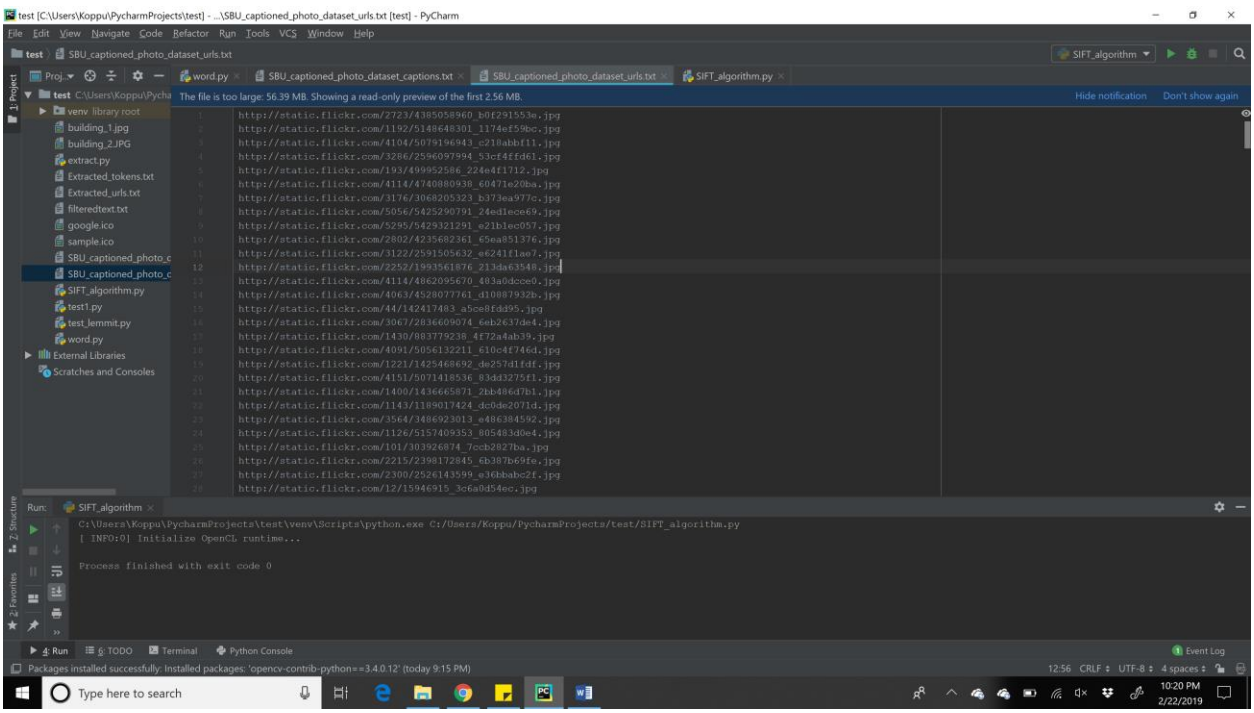
- nltk
- opencv-python
- numpy
- matplotlib
- Tensorflow

Dataset :

Thee are many datasets available out of which an SBU dataset is chosen, which gives the data in the form of two separate files "URL's" and the "captions" respectively.

The following screenshots describes the dataset URL's and their corresponding captions.

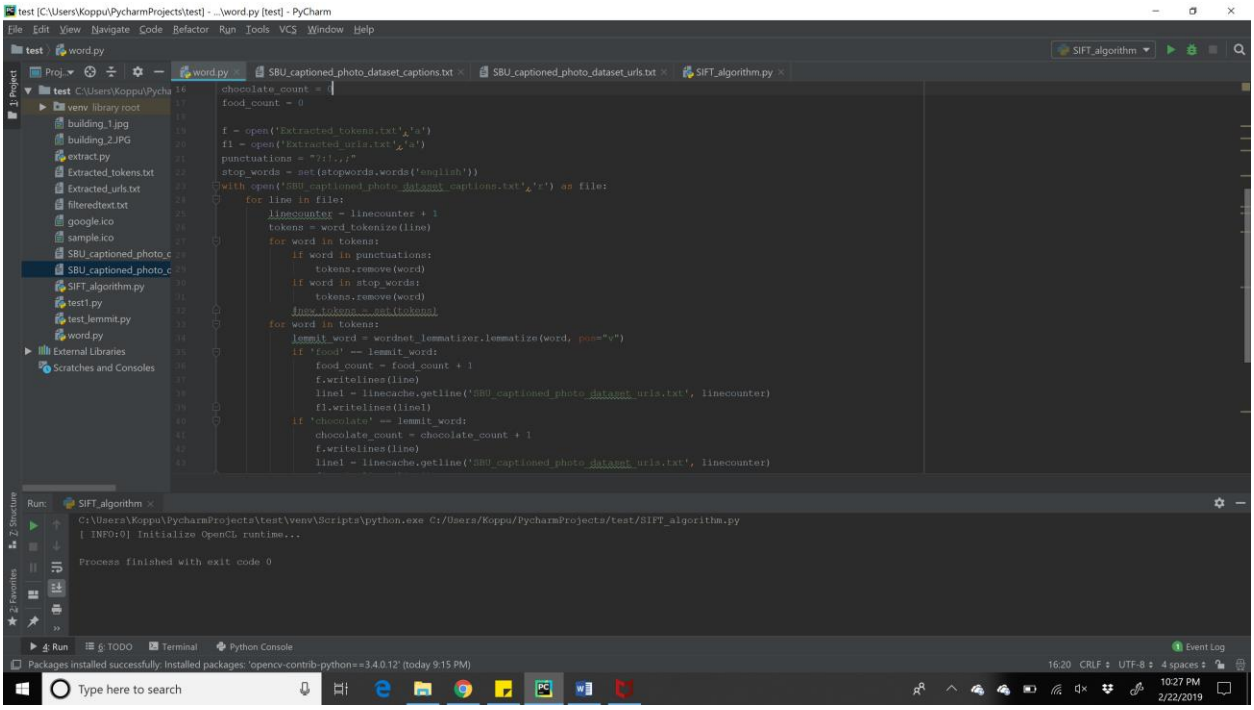




Tokenization and Lemmatization:

- Tokenization generally splits the sentence or a paragraph and splits them into words and store them in the form of list.
- It can be done in two ways namely word tokenization(word_tokenize) and sentence tokenization(sent_tokenize).

- We have used word tokenization in order to extract the different keywords required to extract the data from the entire dataset.
- By tokenizing the data the extracted captions are stored in the new file and by using the line cache the corresponding urls are retrieved and stored in another file.



Lemmatization:

- Lemmatization is the process of getting a root word to the given keyword which is very helpful inoder to pre-process the data.
- For example, consider a word running which is present in the caption. It’s root word is run and when one predicts the output on ‘run’ may loose certain data.
- We eliminate stopwords when lemmitization is done as those words are not important.

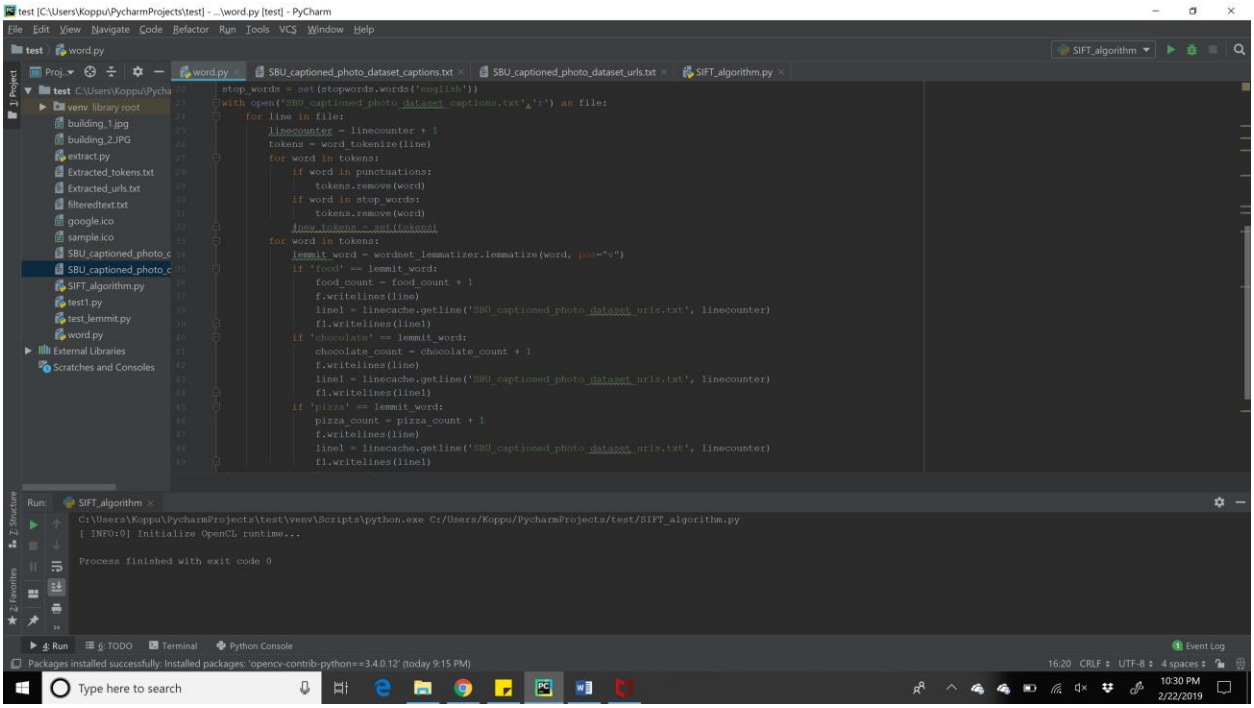
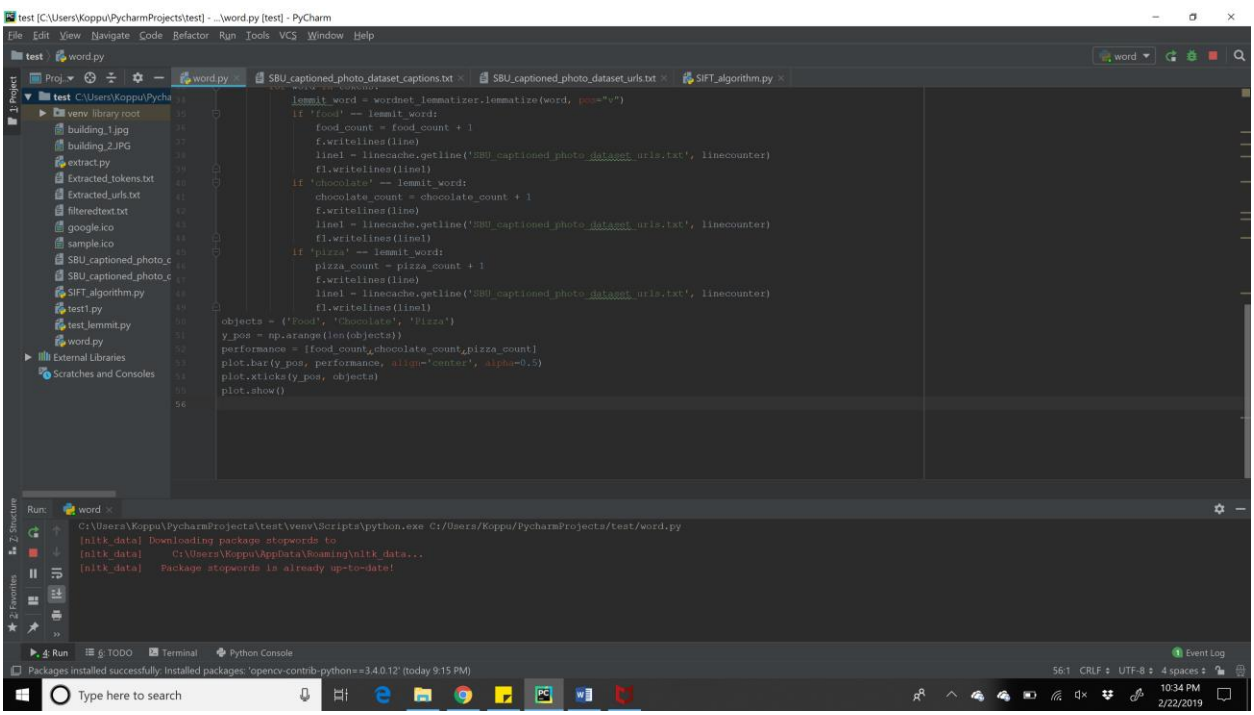
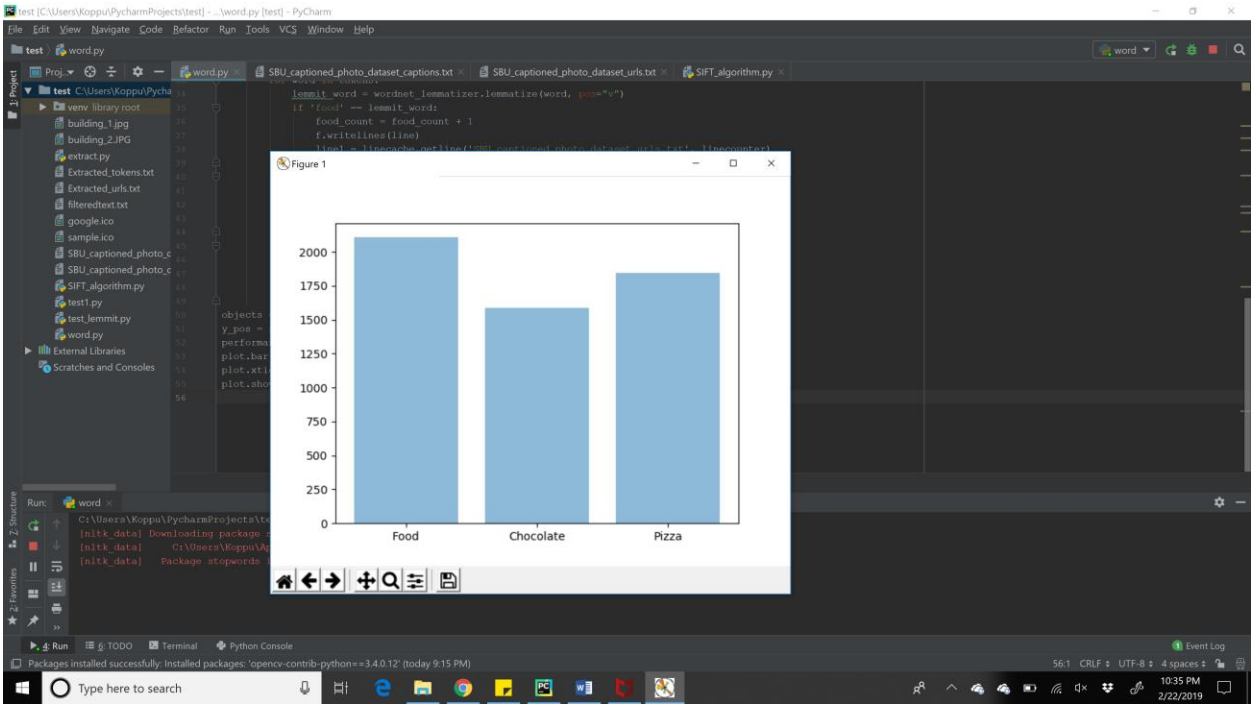


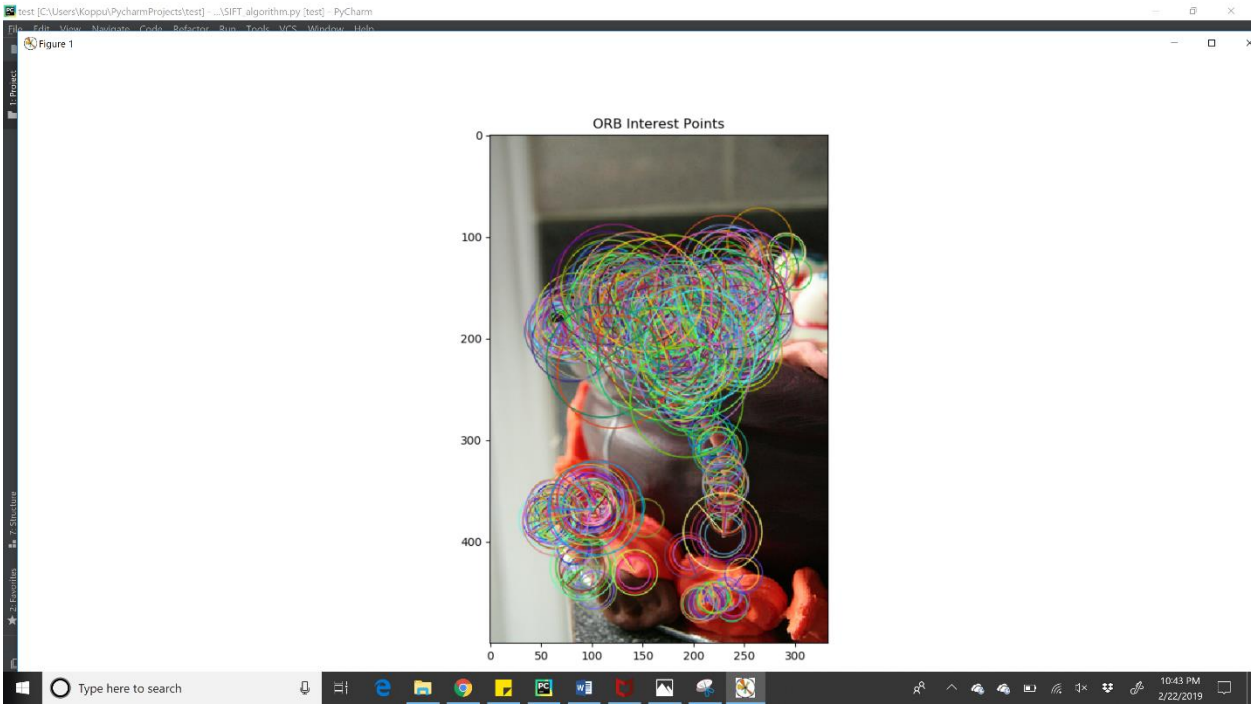
Image statistics:

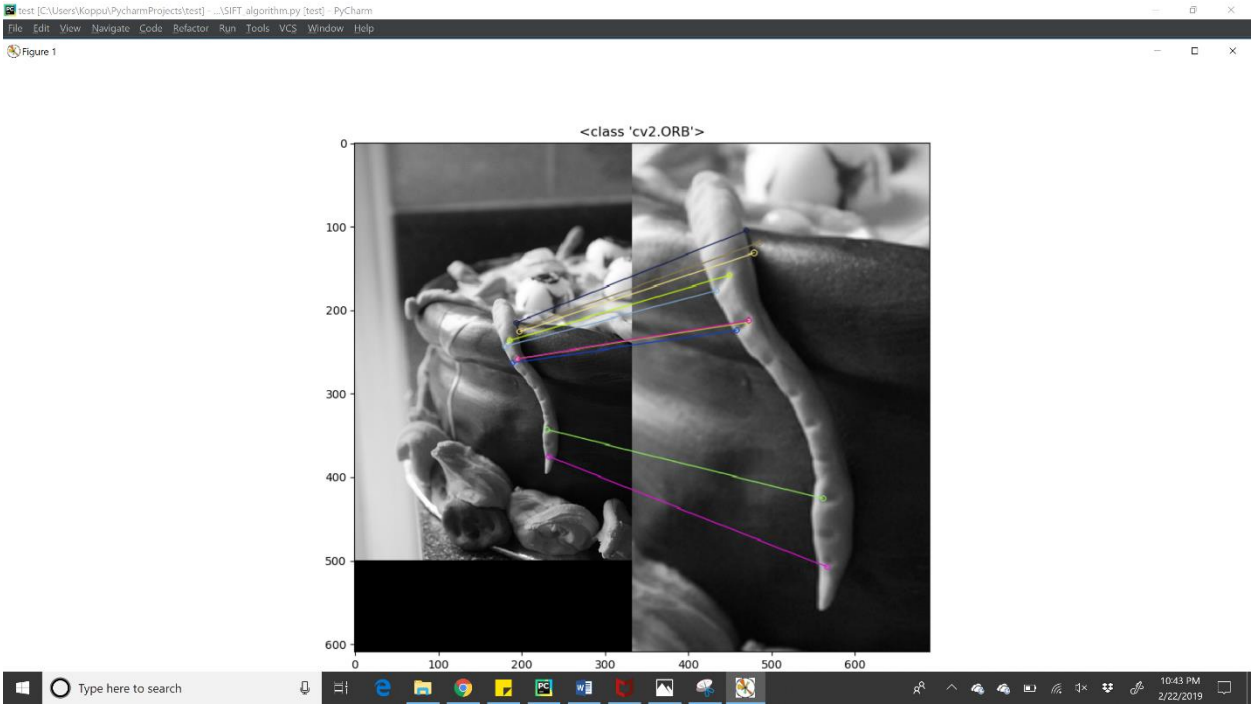
A word count has been performed and respected data has been plot using a matplotlib library.

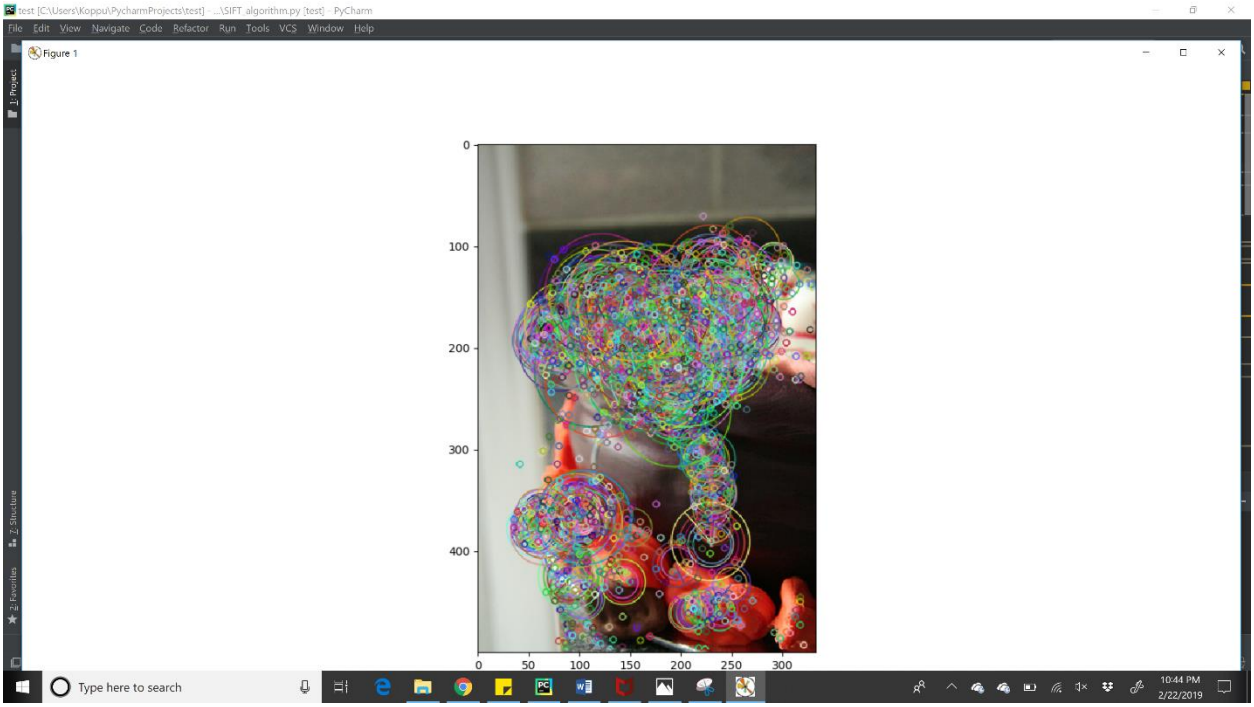


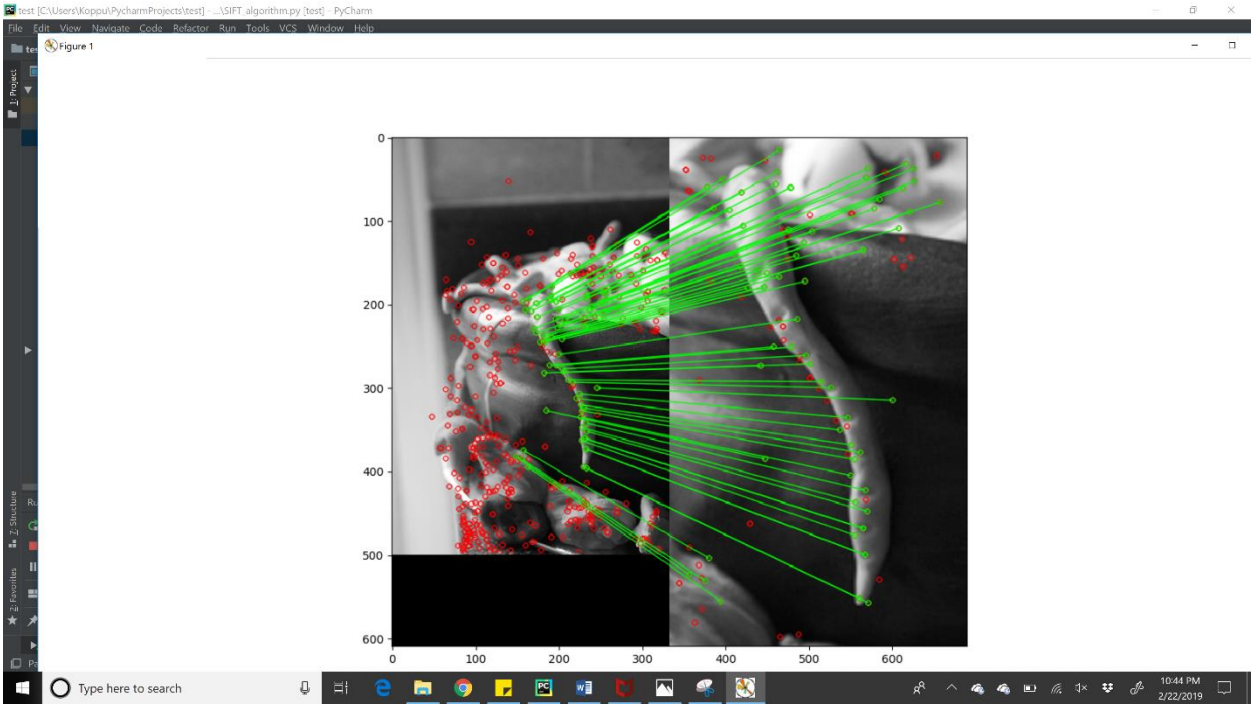


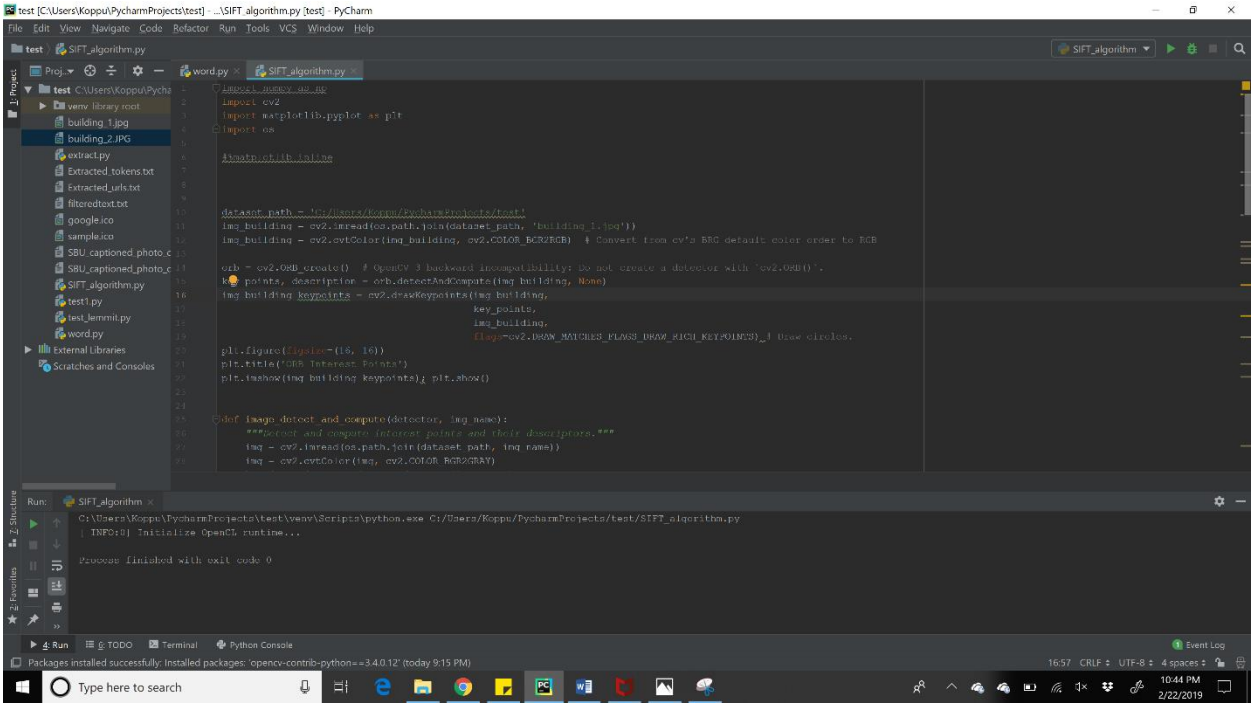
SIFT Algorithm:


test [C:\Users\Koppu\PycharmProjects\test] - ...\SIFT_algorithm.py [test] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Figure 1
ORB Interest Points
0
100
200
300
400
0 50 100 150 200 250 300
Type here to search
10:43 PM
2/22/2019


test [C:\Users\Koppu\PycharmProjects\test] - ...\SIFT_algorithm.py [test] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Figure 1
<class 'cv2.ORB'>
0
100
200
300
400
500
600
0 100 200 300 400 500 600
Type here to search
10:43 PM
2/22/2019


test [C:\Users\Koppu\PycharmProjects\test] - ...\SIFT_algorithm.py [test] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Figure 1
0
100
200
300
400
0 50 100 150 200 250 300
Type here to search
10:44 PM
2/22/2019


test [C:\Users\Koppu\PycharmProjects\test] - ...\SIFT_algorithm.py [test] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Figure 1
0
100
200
300
400
500
600
0 100 200 300 400 500 600
Type here to search
10:44 PM
2/22/2019


test [C:\Users\Koppu\PycharmProjects\test] - ...\SIFT_algorithm.py [test] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
test SIFT_algorithm.py
SIFT_algorithm
word.py
SIFT_algorithm.py
test C:\Users\Koppu\Pycha
venv library root
building_1.jpg
building_2.JPG
extract.py
Extracted_tokens.txt
Extracted_urls.txt
filteredtext.txt
google.ico
sample.ico
SBU_captioned_photo_c
SBU_captioned_photo_c
SIFT_algorithm.py
test1.py
test_lemmit.py
word.py
External Libraries
Scratches and Consoles
import cv2
import matplotlib.pyplot as plt
import os
%matplotlib inline
dataset_path = 'C:/Users/Koppu/PycharmProjects/test'
img_building = cv2.imread(os.path.join(dataset_path, 'building_1.jpg'))
img_building = cv2.cvtColor(img_building, cv2.COLOR_BGR2RGB) # Convert from cv's BRG default color order to RGB
orb = cv2.ORB_create() # OpenCV 3 backward incompatibility: Do not create a detector with `cv2.ORB()`.
key_points, description = orb.detectAndCompute(img_building, None)
img_building_keypoints = cv2.drawKeypoints(img_building,
key_points,
img_building,
flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)# Use circles.
plt.figure(figsize=(16, 16))
plt.title('ORB Interest Points')
plt.imshow(img_building_keypoints); plt.show()
def image_detect_and_compute(detector, img_name):
"""Detect and compute interest points and their descriptors."""
img = cv2.imread(os.path.join(dataset_path, img_name))
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
Run: SIFT_algorithm
C:\Users\Koppu\PycharmProjects\test\venv\Scripts\python.exe C:/Users/Koppu/PycharmProjects/test/SIFT_algorithm.py
[INFO:0] Initialize OpenCL runtime...
Process finished with exit code 0
Run TODO Terminal Python Console
Event Log
Packages installed successfully: Installed packages: 'opencv-contrib-python==3.4.0.12' (today 9:15 PM)
16:57 CRLF UTF-8 4 spaces
Type here to search
10:44 PM
2/22/2019