CS5542 Big Data Analytics and App

Lab Assignment #1

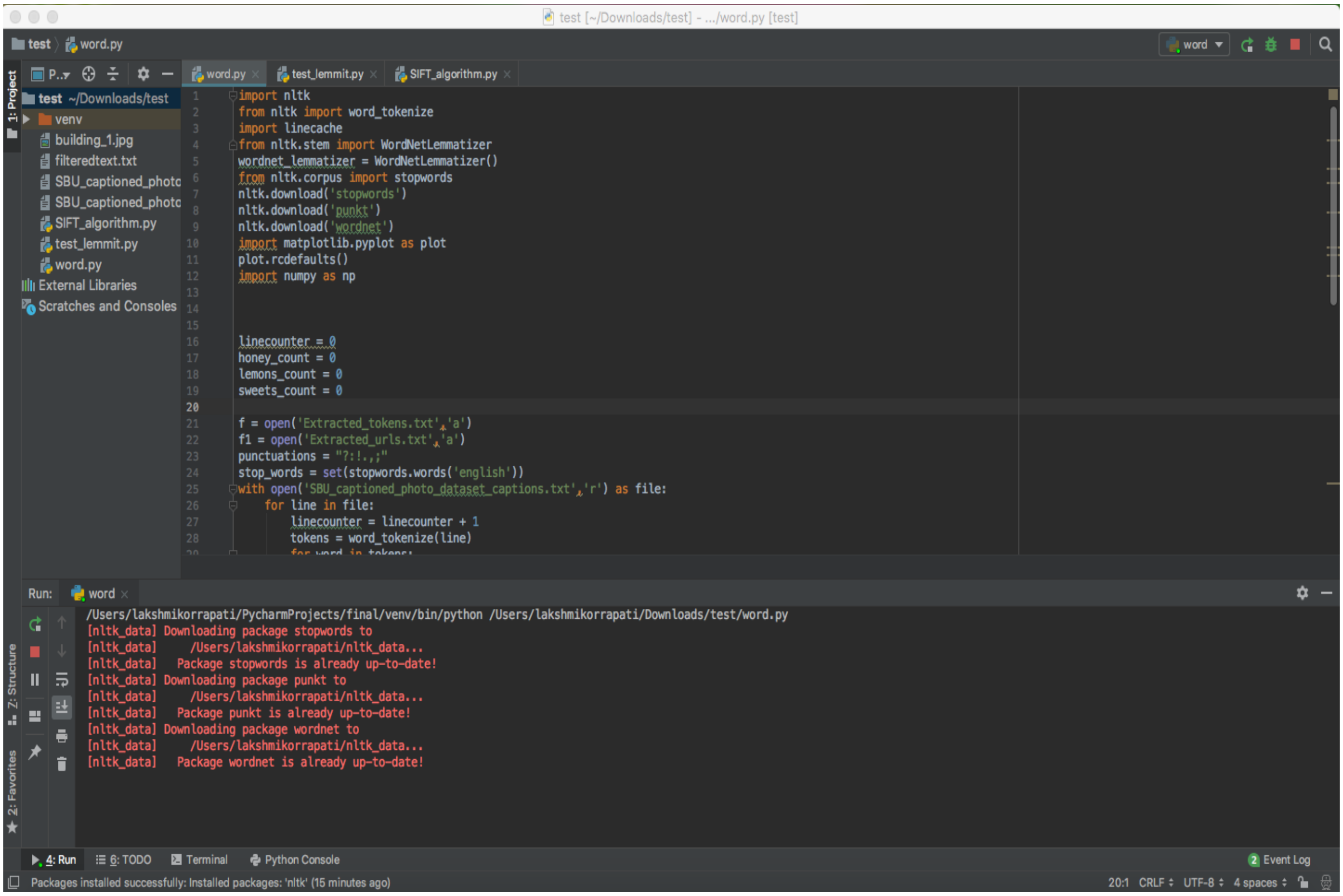
Name: Lakshmi Korrapati

ID: 14

1. NLP Tokenization: Sentence Tokenization otherwise called sentence breaking, is the issue in Natural language processing of choosing where sentences start and end. Regularly devices require their contribution to be partitioned into sentences for various reasons. Taking text and breaking into individual words.

   NLP Lemmatization: Stemming and Lemmatization are the fundamental content handling strategies for English content. The objective of both stemming and lemmatization is to diminish inflectional structures and once in a while derivationally related types of a word to a typical base structure.

Word.py



```python
import nltk
from nltk import word_tokenize
import linecache
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
import matplotlib.pyplot as plot
plot.rcdefaults()
import numpy as np



linecounter = 0
honey_count = 0
lemons_count = 0
sweets_count = 0

f = open('Extracted_tokens.txt','a')
f1 = open('Extracted_urls.txt','a')
punctuations = "?:!.,;"
stop_words = set(stopwords.words('english'))
with open('SBU_captioned_photo_dataset_captions.txt','r') as file:
    for line in file:
        linecounter = linecounter + 1
        tokens = word_tokenize(line)
        for word in tokens:
```

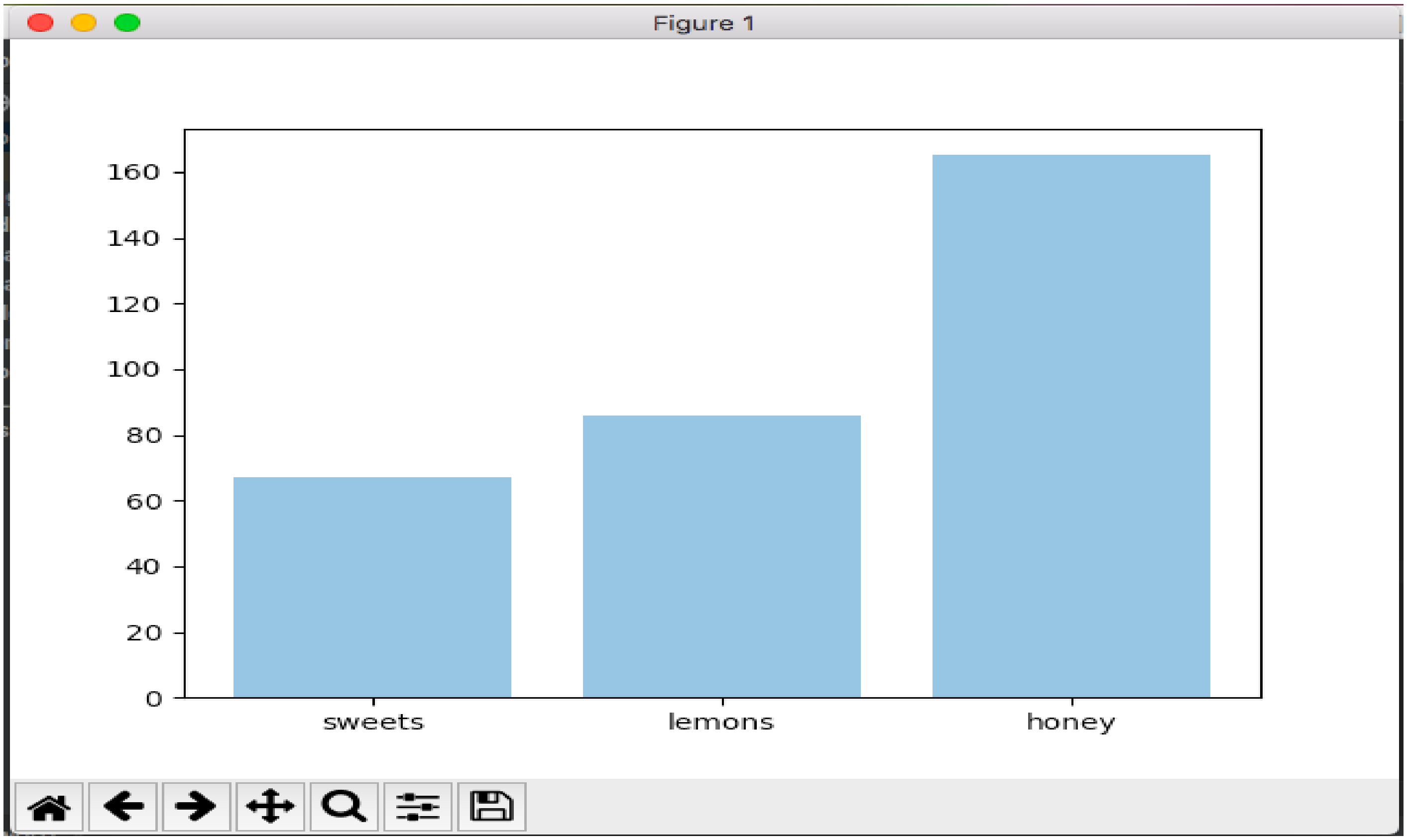
```
/Users/lakshmikorrapati/PycharmProjects/final/venv/bin/python /Users/lakshmikorrapati/Downloads/test/word.py
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/lakshmikorrapati/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/lakshmikorrapati/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/lakshmikorrapati/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Packages installed successfully: Installed packages: 'nltk' (15 minutes ago)

which extracted the urls from given dataset and implemented our own extracted urls from dataset.

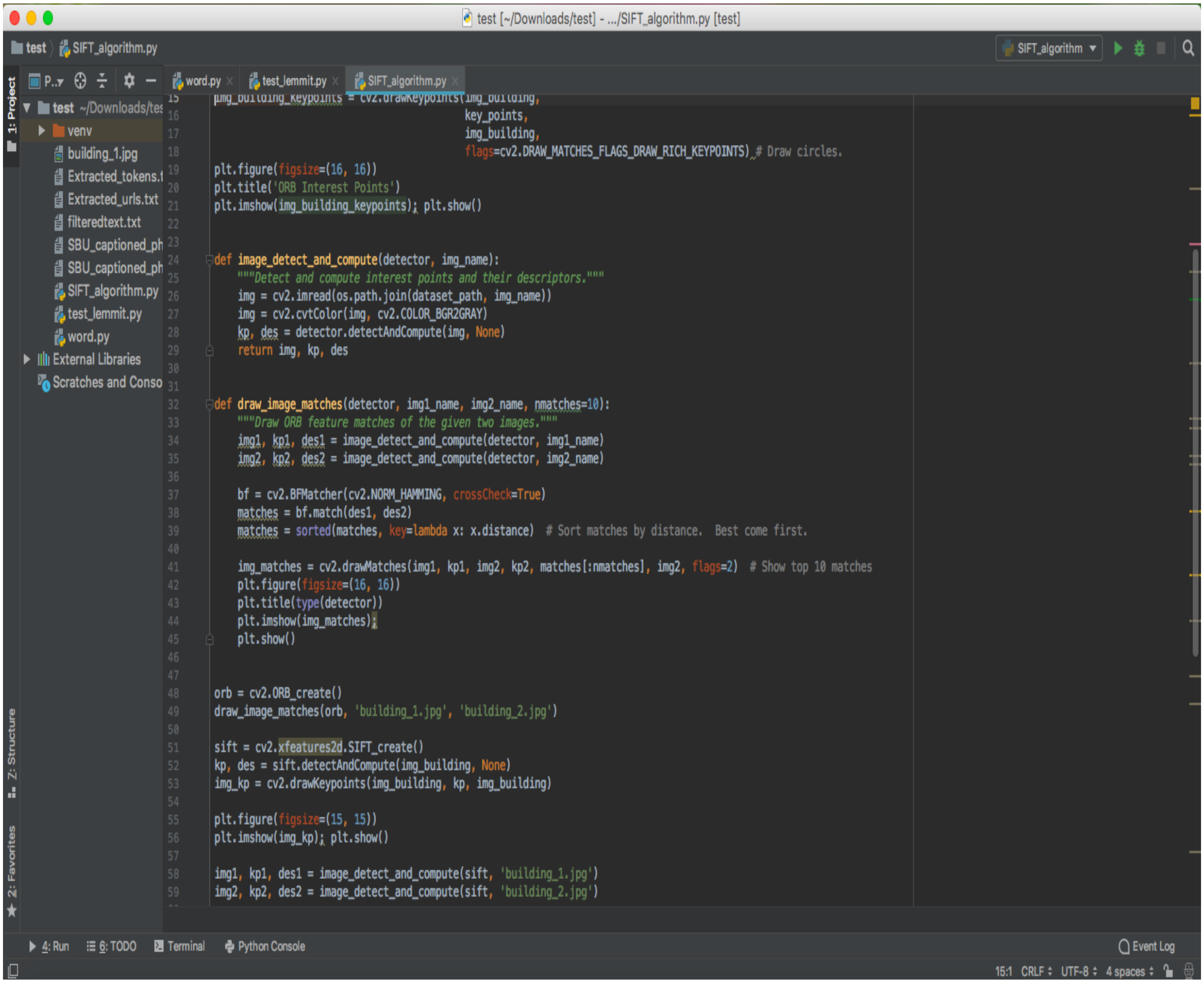
The output id shown below in graph. The dataset given by as three classes sweets, lemons, honey.

Output:



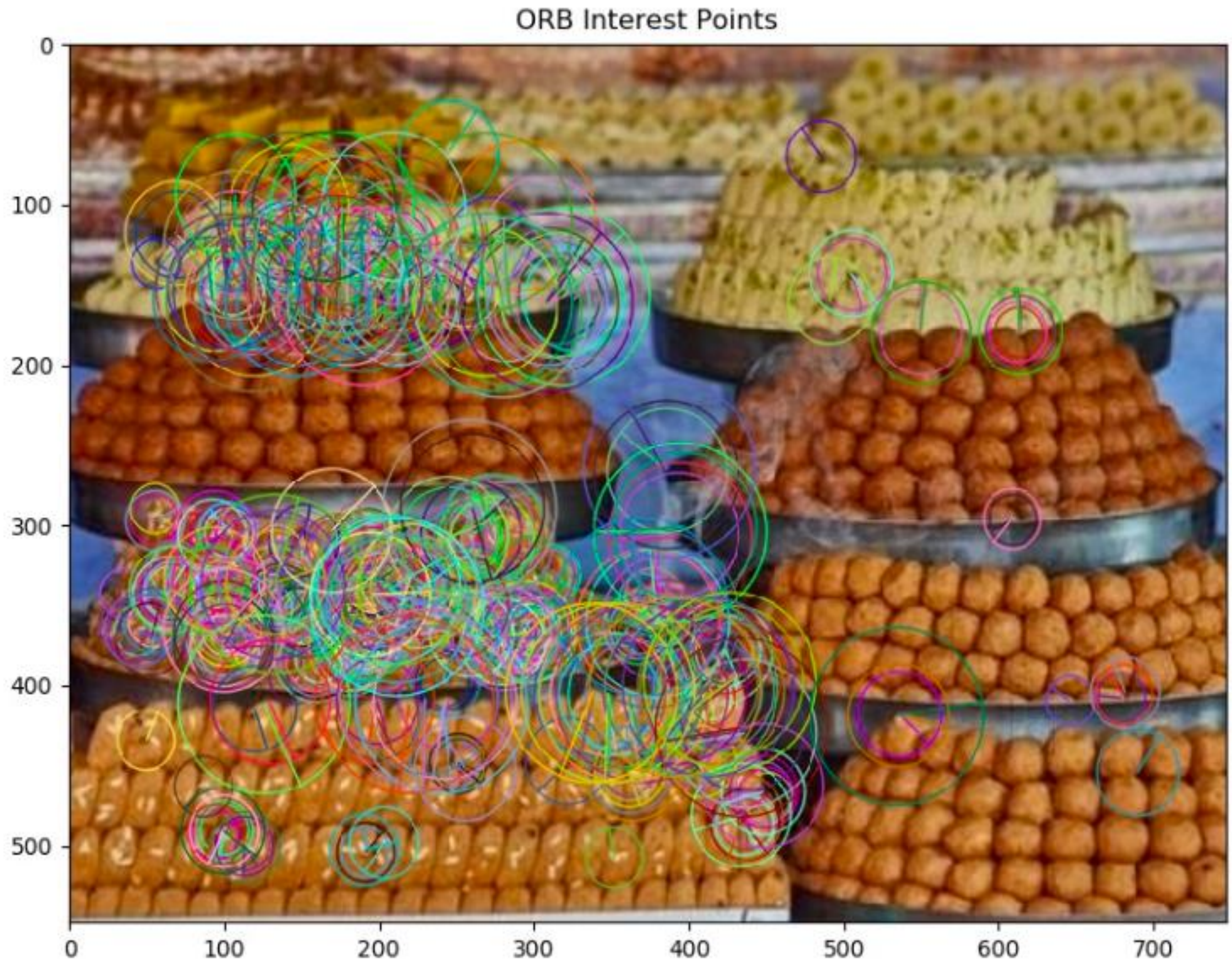
Three classes are taken for image captioning. They are sweets, lemons and honey.

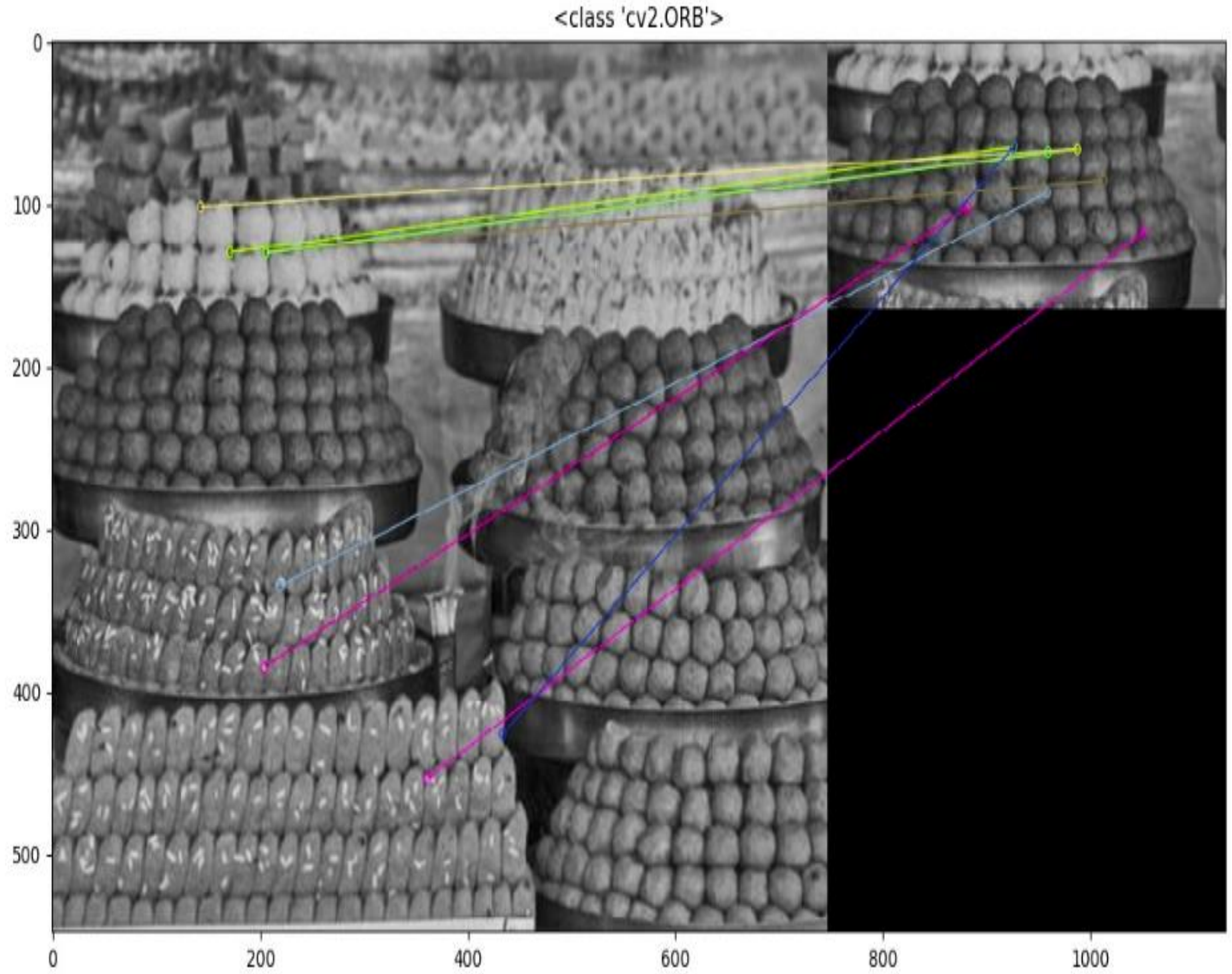
Shift feature extraction:

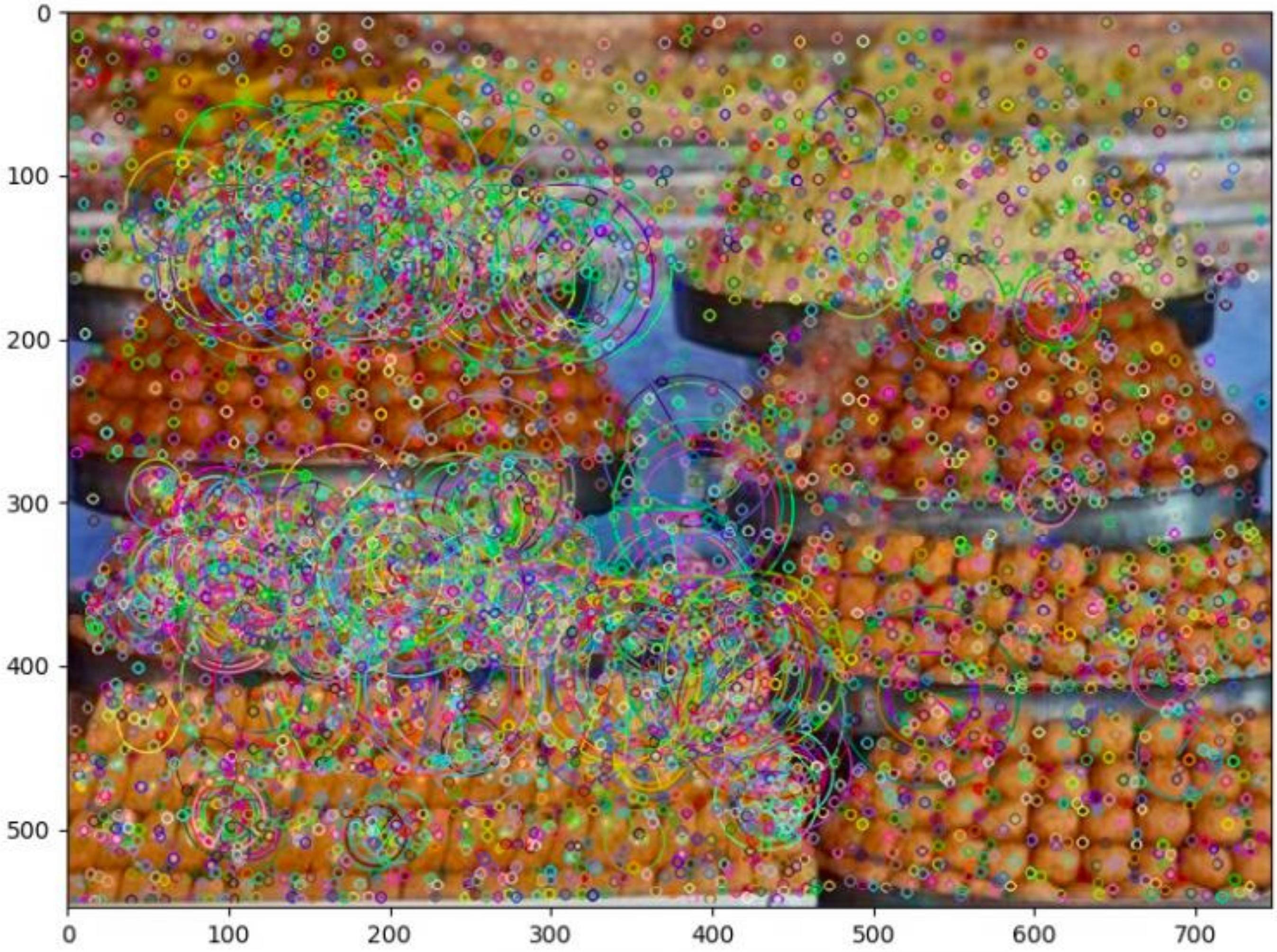


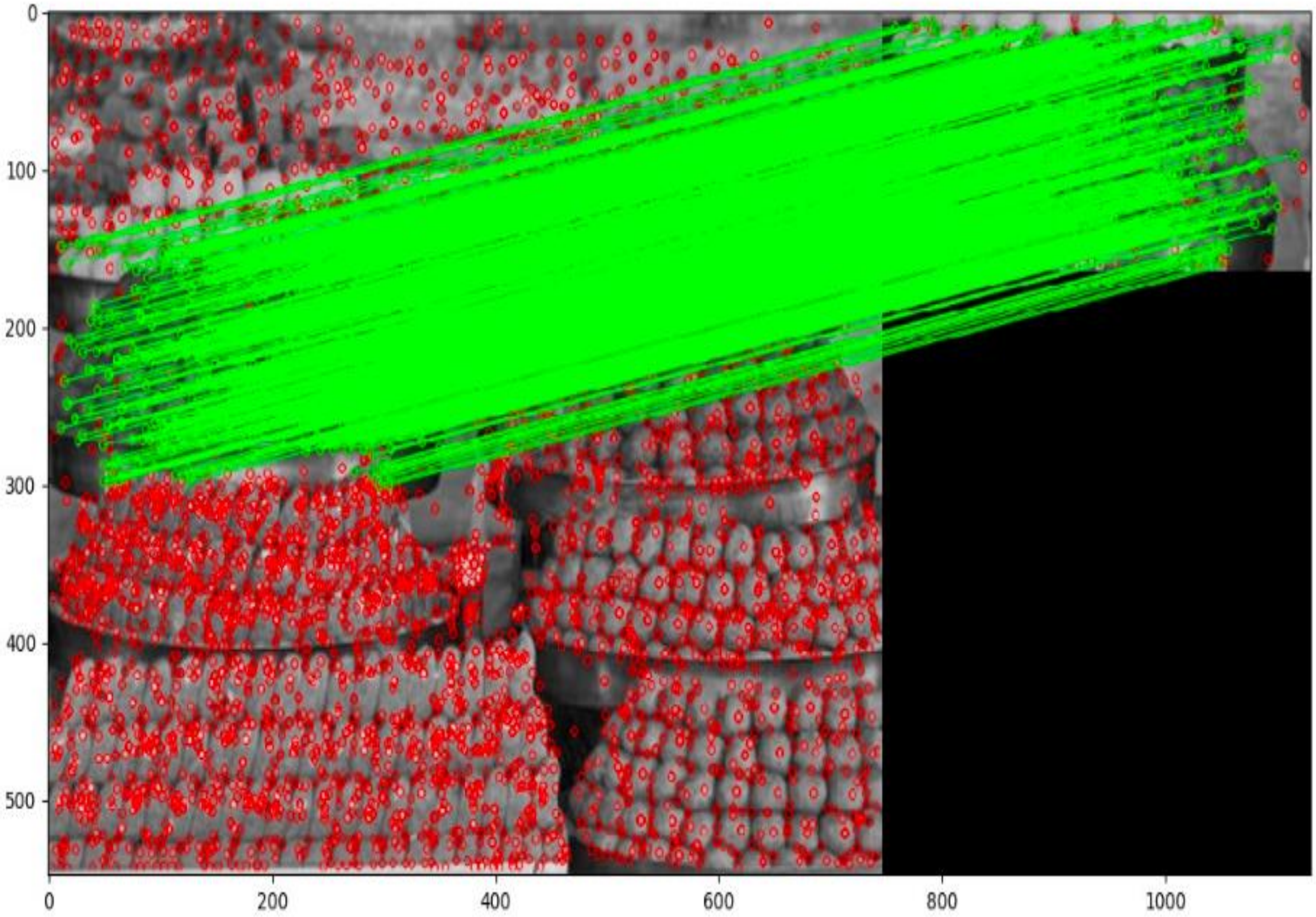
```python
img_building_keypoints = cv2.drawKeypoints(img_building,
                                           key_points,
                                           img_building,
                                           flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS) # Draw circles.
plt.figure(figsize=(16, 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)
    kp, des = detector.detectAndCompute(img, None)
    return img, kp, des


def draw_image_matches(detector, img1_name, img2_name, nmatches=10):
    """Draw ORB feature matches of the given two images."""
    img1, kp1, des1 = image_detect_and_compute(detector, img1_name)
    img2, kp2, des2 = image_detect_and_compute(detector, img2_name)

    bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
    matches = bf.match(des1, des2)
    matches = sorted(matches, key=lambda x: x.distance)  # Sort matches by distance.  Best come first.

    img_matches = cv2.drawMatches(img1, kp1, img2, kp2, matches[:nmatches], img2, flags=2)  # Show top 10 matches
    plt.figure(figsize=(16, 16))
    plt.title(type(detector))
    plt.imshow(img_matches);
    plt.show()


orb = cv2.ORB_create()
draw_image_matches(orb, 'building_1.jpg', 'building_2.jpg')

sift = cv2.xfeatures2d.SIFT_create()
kp, des = sift.detectAndCompute(img_building, None)
img_kp = cv2.drawKeypoints(img_building, kp, img_building)

plt.figure(figsize=(15, 15))
plt.imshow(img_kp); plt.show()

img1, kp1, des1 = image_detect_and_compute(sift, 'building_1.jpg')
img2, kp2, des2 = image_detect_and_compute(sift, 'building_2.jpg')
```

Output: the below images are taken from the class dataset name ‘sweets’. After the programe executed the images features are extracted and displayed.


ORB Interest Points
0
100
200
300
400
500
0
100
200
300
400
500
600
700


<class 'cv2.ORB'>
0
100
200
300
400
500
0
200
400
600
800
1000


0
100
200
300
400
500
0
100
200
300
400
500
600
700


0
100
200
300
400
500
0
200
400
600
800
1000