CENG 483

Introduction to Computer Vision

Fall 2021-2022

Take Home Exam 2

Object Recognition

Student Random ID:

Please fill in the sections below only with the requested information. If you have additional things to mention, you can use the last section. Please note that all of the results in this report should be given for the **validation set**. Also, when you are expected to comment on the effect of a parameter, please make sure to fix other parameters.

1 Local Features (25 pts)

- Explain SIFT and Dense-SIFT in your own words. What is the main difference?

  SIFT is used to extract local features in an image. It is scale invariant as the name suggests. Dense-SIFT extracts more features because it gives sift descriptor on each location. Because of that, it is computationally more expensive compared to SIFT.

- Put your quantitative results (classification accuracy) regarding 5 values of SIFT and 3 values of Dense-SIFT parameters here. In SIFT change each parameter once while keeping others same and in Dense-SIFT change size of feature extraction region. Discuss the effect of these parameters by using 128 clusters in k-means and 8 nearest neighbors for classification.
  I tried changing default parameters as follows:

  nfeatures = 25 – Accuracy = %20.54
  nOctaveLayers= 12 – Accuracy = %21.02
  contrastThreshold= 0.08 – Accuracy = %18.54
  edgeThreshold= 20 – Accuracy = %20.42
  sigma= 10 – Accuracy = %12.22

  Best configuration is to use nOctaveLayers = 12

2 Bag of Features (45 pts)

- How did you implement BoF? Briefly explain.
  After clustering all the local features using kmeans, I iterate through the validation images and for every type of image (apple, pear, etc) I do predictions according to cluster centers and get histogram of these predictions. After that, for every type I add histograms to the corresponding list and when

iteration over the same type is over, I add all these histograms to the corresponding key of BoW. At the end I end up with all the histograms and which class they correspond to.

- Give pseudo-code for obtaining the dictionary.

Algorithm 1 Obtaining Dictionary

```
for img in imgList do
    detectors ← get_sift_detectors(img)
    for detector in detectors do
        vocab+ = detector
```

- Give pseudo-code for obtaining BoF representation of an image once the dictionary is formed.

Algorithm 2 Obtaining BoF representation of an image

```
detectors ← get_sift_detectors(img)
BoF_rep ← kmeans.predict(detectors) =0
```

- Put your quantitative results (classification accuracy) regarding 3 different parameter configurations for the BoF pipeline here. Discuss possible reasons for each one's relatively better/worse accuracy. You are suggested to keep $k \leq 1024$ in k-means to keep experiment durations managable. You need to use the best feature extractor you obtained in the previous part together with the same classifier.

  I tried k values of 128, 256 and 512 but obtained same number of correct guesses and accuracy and it is %21.89

3 Classification (30 pts)

- Put your quantitative results regarding k-Nearest Neighbor Classifier for k values 16, 32 and 64 by using the best k-means representation and feature extractor. Discuss the effect of these briefly.
- What is the accuracy values, and how do you evaluate it? Briefly explain.

  Since k value of kmeans gave same results I will stick with the k value of 128 for the kmeans. Results using SIFT with nOctaveLayers = 12 are as following:
  knn value = 16 – Accuracy = %21.89
  knn value = 32 – Accuracy = %24.76
  knn value = 64 – Accuracy = %23.09

- Give confusion matrices for classification results of these combinations.
  knn value = 16:
  [39, 2, 3, 7, 3, 1, 4, 15, 10, 2, 5, 1, 3, 3, 2],
  [8, 3, 0, 11, 6, 2, 8, 10, 10, 5, 5, 3, 9, 12, 8],
  [7, 3, 14, 8, 3, 1, 3, 14, 8, 3, 9, 5, 3, 17, 1],
  [4, 5, 3, 21, 8, 3, 11, 18, 5, 1, 1, 1, 6, 3, 10],
  [7, 2, 1, 10, 14, 6, 10, 6, 7, 0, 13, 1, 2, 15, 6],

[9, 3, 4, 7, 5, 5, 6, 9, 10, 9, 9, 6, 5, 10, 3],
[11, 3, 5, 14, 7, 0, 10, 11, 11, 4, 5, 4, 3, 10, 2],
[18, 2, 3, 8, 4, 2, 4, 45, 2, 3, 2, 4, 0, 2, 1],
[3, 2, 2, 2, 4, 1, 2, 9, 55, 6, 3, 3, 1, 6, 1],
[9, 0, 3, 6, 1, 0, 4, 13, 32, 17, 7, 2, 2, 2, 1],
[9, 2, 8, 4, 2, 1, 3, 9, 9, 5, 27, 5, 5, 9, 2],
[12, 5, 4, 7, 1, 2, 8, 11, 7, 1, 11, 13, 2, 14, 2],
[17, 5, 5, 5, 0, 1, 2, 13, 11, 2, 12, 3, 5, 15, 4],
[2, 3, 2, 6, 3, 1, 1, 6, 5, 2, 12, 1, 1, 54, 1],
[6, 4, 0, 14, 6, 2, 10, 13, 6, 6, 10, 2, 5, 10, 6]

knn value = 32:
[50, 3, 4, 7, 2, 0, 5, 11, 10, 2, 2, 0, 0, 2, 2],
[10, 4, 1, 7, 3, 6, 7, 11, 13, 7, 4, 2, 6, 14, 5],
[8, 1, 18, 8, 5, 0, 6, 13, 8, 1, 8, 3, 2, 17, 1],
[7, 4, 3, 23, 10, 2, 10, 19, 4, 3, 3, 0, 1, 6, 5],
[8, 5, 0, 11, 11, 6, 4, 5, 10, 1, 13, 0, 4, 17, 5],
[9, 3, 2, 10, 6, 2, 4, 13, 16, 7, 8, 5, 4, 10, 1],
[13, 0, 3, 13, 6, 1, 8, 15, 12, 5, 3, 3, 3, 11, 4],
[17, 0, 5, 9, 4, 2, 3, 47, 3, 2, 0, 3, 0, 3, 2],
[5, 1, 2, 2, 1, 1, 3, 8, 58, 8, 4, 2, 0, 4, 1],
[9, 0, 4, 3, 2, 1, 3, 11, 29, 20, 8, 1, 2, 3, 3],
[15, 2, 8, 4, 3, 1, 1, 9, 11, 6, 24, 4, 1, 11, 0],
[13, 2, 5, 12, 2, 0, 2, 10, 7, 1, 11, 20, 1, 12, 2],
[15, 2, 3, 3, 2, 2, 2, 18, 14, 4, 8, 4, 4, 17, 2],
[1, 1, 4, 2, 3, 1, 3, 6, 5, 2, 14, 1, 2, 54, 1],
[13, 5, 1, 15, 4, 0, 5, 13, 8, 2, 11, 1, 4, 13, 5]

knn value = 64:
[49, 3, 3, 9, 0, 0, 4, 13, 10, 0, 2, 0, 1, 5, 1],
[10, 6, 1, 7, 9, 1, 5, 10, 13, 6, 4, 2, 4, 15, 7],
[9, 0, 14, 10, 1, 0, 4, 10, 5, 4, 12, 1, 5, 22, 2],
[8, 5, 2, 20, 12, 4, 7, 18, 3, 3, 1, 0, 2, 8, 7],
[5, 3, 0, 7, 18, 4, 5, 13, 11, 1, 11, 0, 6, 14, 2],
[10, 4, 3, 12, 3, 1, 4, 11, 21, 6, 6, 3, 3, 12, 1],
[15, 1, 2, 10, 6, 1, 13, 18, 10, 4, 3, 2, 3, 12, 0],
[26, 0, 2, 7, 0, 2, 0, 50, 5, 3, 0, 0, 0, 3, 2],
[7, 1, 2, 5, 1, 0, 4, 8, 57, 5, 3, 0, 2, 5, 0],
[8, 0, 1, 4, 2, 2, 1, 13, 35, 15, 6, 2, 1, 6, 3],
[15, 3, 5, 3, 2, 3, 3, 10, 14, 4, 23, 3, 2, 9, 1],
[13, 3, 2, 9, 1, 0, 1, 11, 8, 2, 12, 15, 3, 18, 2],
[18, 2, 2, 5, 0, 2, 3, 17, 12, 4, 9, 3, 4, 18, 1],
[1, 3, 3, 3, 3, 0, 4, 4, 5, 2, 12, 0, 0, 58, 2],
[11, 8, 1, 12, 6, 1, 7, 16, 4, 5, 10, 1, 2, 13, 3]

4 Additional Comments and References

I couldn't implement the dense-SIFT so all my results reported according to only SIFT values.