UNIVERSITÀ DI PISA

A.Y. 2019/20
« Intelligent systems for pattern recognition »
Master Degree in Computer Science
Artificial Intelligence Curriculum

Midterm 1
Assignment 5

Image processing with SIFT

Diletta Goglia

1. Sift detector and descriptor

Detecting some relevant points by using SIFT detector and placing a SIFT descriptor on each of those point, in order to obtain a bag of SIFT descriptors (i.e. intensity gradient histograms) to represent each image (for a total of 8 images, belonging to different thematic subset).

```
import cv2
import glob #module that finds all the pathnames matching
from matplotlib import pyplot as plt # importing library for plotting

#load images
collection = []
for im in glob.glob('selected_images/*.bmp'):
    img = cv2.imread(im)
    collection.append(img)
#convert them into gray scale
image_list=[]
for i in collection:
    gray = cv2.cvtColor(i, cv2.COLOR_BGR2GRAY)
    image_list.append(gray)

#create feature extraction object
sift = cv2.xfeatures2d.SIFT_create() #load algorythm
# optional: passing parameters like numbers of features and threshold
# install opencv-contrib-python to make the attribute works

# Detect: find relevant keypoints
kp = [] #list of keypoints
for img in image_list:
    k = sift.detect(img, None) # passing no mask
    kp.append(k)

des = [] #list of descriptors
for n in range(len(image_list)):
    des.append(n)
    #Compute a descriptor for each keypoint detected
    kp[n], des[n] = sift.compute(image_list[n], kp[n])
    #output: keypoint and corresponding descriptor --> MATRIX(rows=kp, col=128)

    # Draw keypoints detected on the image
    image_list[n] = cv2.drawKeypoints(image_list[n], kp[n],
                        None, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
    #the flag draw a circle with size of keypoint and show its orientation

#for each image, I got a number of descriptors equal to the number of keypoints
```

2. Histogram comparison

A. Visual comparison:

plotting the two SIFT descriptors closeby as barplots

```
images = []
for im in glob.glob('result/*.bmp'):
    image = cv2.imread(im)
    images.append(image)

#creating multiple plots
fig = plt.figure(figsize=(20, 20)) # create a figure object

ax1 = plt.subplot(321) # face
ax2 = plt.subplot(322) # horse

...

#pick random descriptors (keypoints) from each histogram
v1 = randint(0, len(kp[5])-1)
face_hist = des[5][v1] # face histogram
v2 = randint(0, len(kp[6])-1)
car_hist = des[6][v2] # car histogram
v3 = randint(0, len(kp[0])-1)
horse_hist = des[0][v3] # horse histogram
v4 = randint(0, len(kp[4])-1)
cow_hist = des[4][v4] # cow histogram

#plot descriptor for each keypoint
ax1.plot(face_hist, 'g-')
ax3.plot(car_hist, 'b-')

...

#comparison (without overlapping bars)
ax5.hist((face_hist, car_hist), bins=10, color=('b', 'g'), alpha=0.5)
ax6.hist((horse_hist, cow_hist), bins=10, color=('r', 'y'), alpha=0.5)

#save plot
plt.savefig("plot.png")
```

B. Quantitative comparison:

three different histograms comparison metrics

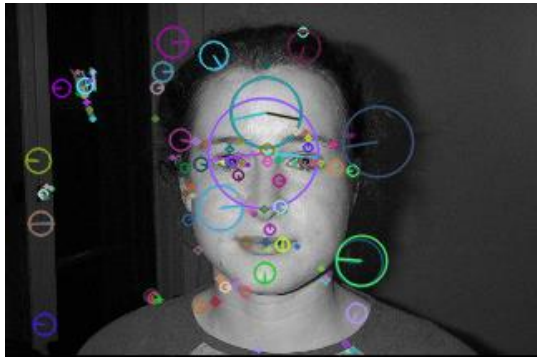
```
#Correlation and Chi-Square
for method in range(3):
    comparison1 = cv2.compareHist(horse_hist, horse_hist, method)
    comp1.append(comparison1)
    comparison2 = cv2.compareHist(horse_hist, car_hist, method)
    comp2.append(comparison2)
    comparison3 = cv2.compareHist(horse_hist, cow_hist, method)
    comp3.append(comparison3)


methods = ["Correlation ", "Chi-Square  ", "Intersection"]

print('\n Method: ' + '\t\t' + 'Horse - Horse ' + '\t' + 'Horse - Car '
      + '\t' + 'Horse - Cow' + '\n')

for c1, c2, c3, m in zip(comp1, comp2, comp3, methods):
    print(m + '\t' + str(c1) + '\t' + str(c2) + '\t' + str(c3) + '\n')
```

```
#to perform Intersection method, histograms have to be normalized
cv2.normalize(horse_hist, horse_hist, alpha=1, norm_type=cv2.NORM_L1)
cv2.normalize(car_hist, car_hist, alpha=1, norm_type=cv2.NORM_L1)
cv2.normalize(cow_hist, cow_hist, alpha=1, norm_type=cv2.NORM_L1)
```

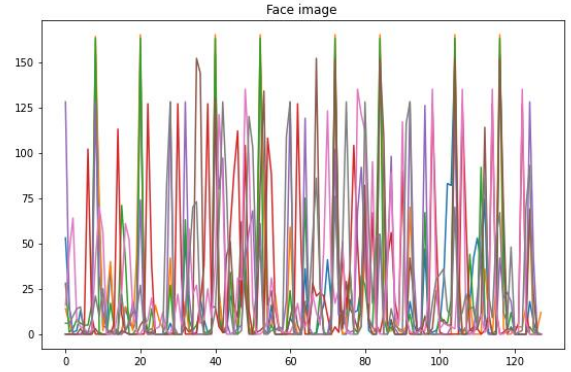
Results

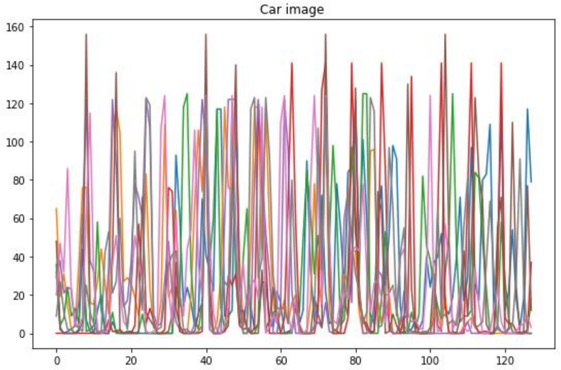
1. Sift detector and descriptor

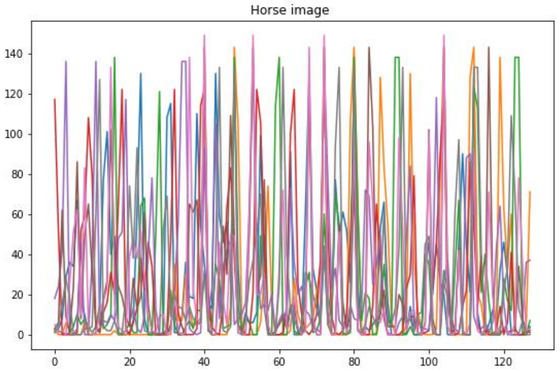
121 keypoints	461 keypoints	412 keypoints	230 keypoints	Keypoints detected

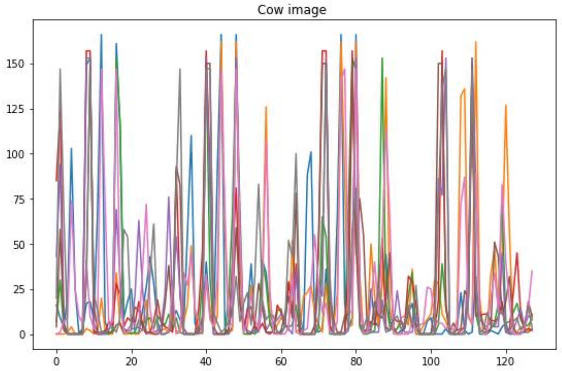



Image









SIFT descriptor

Results

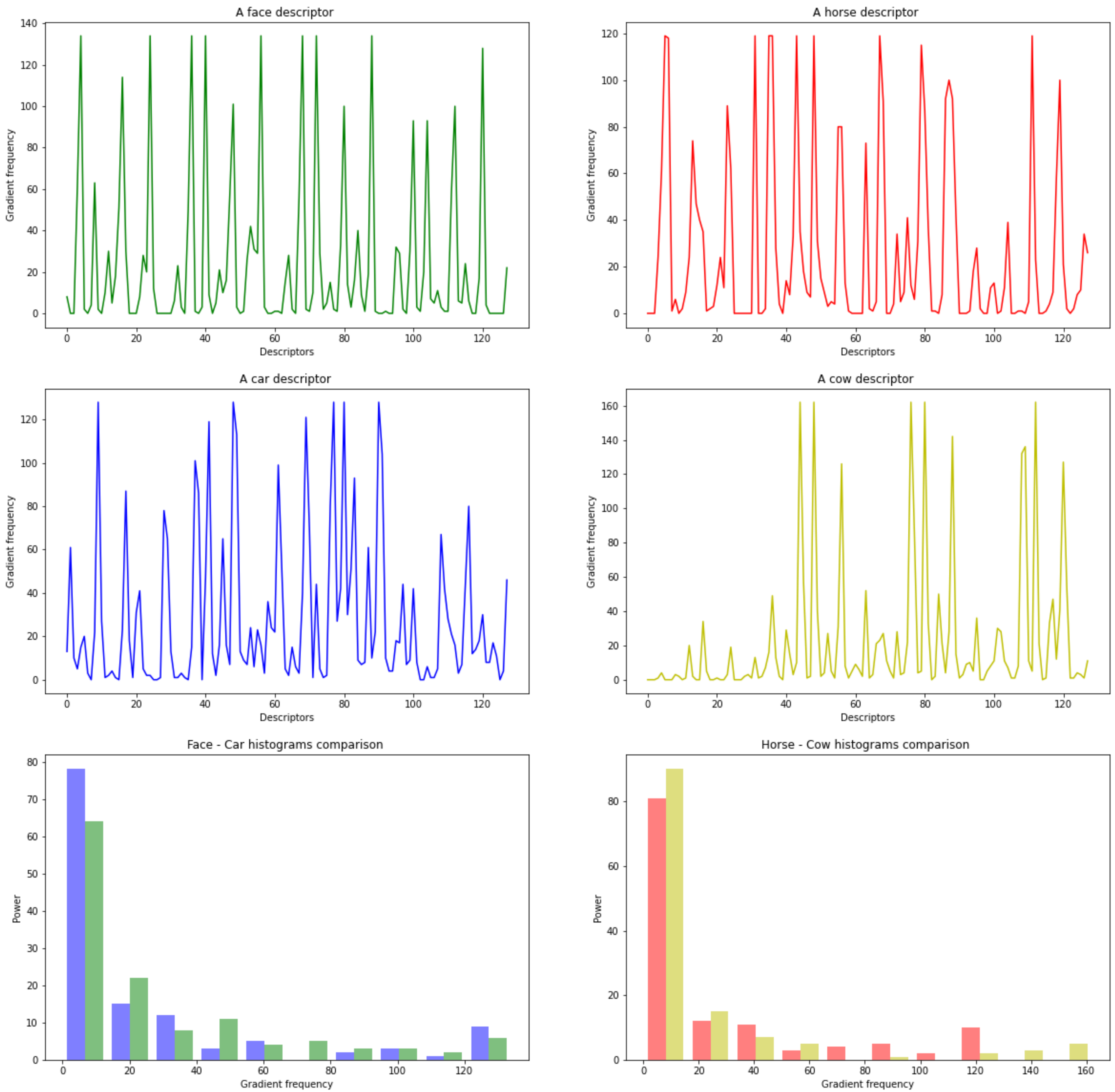
2. Histogram comparison

A. Visual comparison

Comparison between two random SIFT descriptors showing completely different informations.
Histograms are compared in pairs (vertically).

The spectral domain reveals interesting information about data:

- Barplots show how many times a frequency is represented in the signal (it's a visual framework of the power of each frequency)
- Each random descriptor show the same trend in the barplot: lower frequencies are always the most represented in the signal
- However we lose the information about which gradient had that specific frequency



2. Histogram comparison

B. Quantitative comparison

The function `compareHist` returns a numeric parameter that express how well two histograms match with each other.

Three different metrics are used:

- ***Correlation***

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}}$$

- ***Chi-Square***

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

- ***Intersection***

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

Results

Method	Horse - Horse	Horse - Car	Horse - Cow
Correlation	1.0	-0.05	0.28
Chi-square	0.0	46875.43	34101.63
Intersection	1.0	0.32	0.49

A descriptor from a horse image is chosen randomly and is compared with two others random-chosen descriptors, one from a car image and one from a cow image.

For ***correlation***, a high score represents a better match than a low score. A perfect match is 1.0 and a maximal mismatch is -1. A value of 0 indicates no correlation (random association).

For ***chi-square***, a low score represents a better match than a high score. A perfect match is 0 and a total mismatch is unbounded (depending on the size of the histogram)

For ***intersection***, a high score represents a better match than a low score. If both histograms are normalized to 1, then a perfect match is 1 and a total mismatch is 0.

Final observations

Visual comparison

Interesting results

- Lower frequencies are the most represented for each signal

Weak aspects

- In the spectral domain we loose some useful informations (which gradient had that specific frequency?)

Quantitative comparison

Interesting results

- Resulting values proves that images are actually different, but…
- Descriptors (chosen randomly in the respective images) seem to show a representative behaviour of the overall image: (a horse seems to be more similar to a cow than a car).

Weak aspects

- How reliable is this evidence? Is an histogram randomly selected into a bag of descriptors really a representative sample?

UNIVERSITÀ DI PISA

A.Y. 2019/20
« Intelligent systems for pattern recognition »
Master Degree in Computer Science
Artificial Intelligence Curriculum

Thanks for your attention