Lehrstuhl Computergrafik
Prof. Dr.-Ing. Hendrik P.A. Lensch
Prof. Dr.-Ing. Andreas Geiger

Eberhard Karls Universität Tübingen

Machine Learning in Graphics & Vision

Exercise 1

Release date: Fri, 24. April 2020 - **Deadline for Homework: Wed, 6. May 2020 - 21:00**

Excercises

1.1 Feature Matching (3+3+4 Points)

Given a dataset $\mathcal{X}$ containing N vectors $\mathbf{x} \in \mathbb{R}^D$, a critical component in computer vision algorithms is finding the nearest neighbor $\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{q}\|_2$ for a query $\mathbf{q} \in \mathbb{R}^D$, *i.e.* a vector $\mathbf{x}^\star \in \mathcal{X}$ with smallest Euclidean distance to $\mathbf{q}$.

a) Randomly generate $N = 2^{10}$ points for a synthetic $\mathcal{X}$ and implement an exhaustive search using NumPy. Benchmark the query time using your implementation for different $D = 1, 11, 21, \dots 491$ and plot the results. Take all elements from $\mathcal{X}$ as a query. What is the complexity of this method?

b) A typical frame from a full-HD video can be represented by 20000 features-vectors of dimensionality $D = 128$. How long would it take to find all matchings of these vectors between two different 2 minute long videos (30 FPS) using `exhaustive_search`? Assume linear dependency in N to find a lower bound.

c) Finding the approx. nearest neighbor can be accelerated by using a KD-tree. Complete the provided implementation and re-run the experiments from a) including plotting the results. Discuss your findings.

1.2 Fashion-MNIST (6+4 Points)

Figure 1: A subset of the Fashion-MNIST dataset with classes “T-shirt/top”, “Trouser”, “Pullover”, “Dress”, “Coat”, “Sandal”, “Shirt”, “Sneaker”, “Bag ” and “Ankle boot”.

We consider the Fashion-MNIST dataset consisting of 60’000 example pairs (image/label) for training (see Figure 1) and 10’000 pairs for testing.
The dataset can be downloaded from https://github.com/zalandoresearch/fashion-mnist and loaded in Python using

```python
def load_mnist(path, kind='train', each=1):
    import os
    import gzip
    import numpy as np

    labels_path = os.path.join(path, '%s-labels-idx1-ubyte.gz'% kind)
    images_path = os.path.join(path, '%s-images-idx3-ubyte.gz'% kind)

    with gzip.open(labels_path, 'rb') as lbpath:
        labels = np.frombuffer(lbpath.read(), dtype=np.uint8, offset=8)

    with gzip.open(images_path, 'rb') as imgpath:
        images = np.frombuffer(imgpath.read(), dtype=np.uint8,
                               offset=16).reshape(len(labels), 784)

    images = images[::each, :]
    labels = labels[::each]

    return images, labels
```

a) Change the implemented KD-tree from the previous task to return the K nearest neighbors in the training data for each test image. The top-K accuracy is the fraction of test images for which the correct label is among the labels obtained by one of the K nearest neighbors. What is the top-K accuracy for $K = 1, 2, \dots, 10$ using the KD-tree?

b) Besides accuracy there exists some other metrics, *e.g. precision* and *recall* defined as

$$\text{precision} = \frac{T_p}{T_p + F_p} \quad \text{and} \quad \text{recall} = \frac{T_p}{T_p + F_n}$$

for true-positives T_p, false positives F_p and false negatives F_n. Consider the 1-nearest-neighbor classifier for two classes “Pullover” (2) and “Shirt” (6). Compute the precision and recall for a 1-nearest-neighbor binary classifier (using KD-trees) between these two categories.

HINT: You are allowed to skip every but each 10-th data point from the train and test set to obtain a reasonable computation time (use `each=10`).