National University of Sciences and Technology
School of Electrical Engineering and Computer Science
Department of Computer Science

CS867: Computer Vision

Fall 2019

Assignment 2

Image features Detection and Matching

Image Classification

Announcement Date: 16th Oct, 2019

Due Date: 24th Oct 2019 at 11:55 pm (on LMS)

Instructor: Dr. Muhammad Moazam Fraz

Task-1 : Image Feature Extraction and Matching

A local image feature is a tiny patch in the image that's invariant to image scaling, rotation and change in illumination. It's like the tip of a tower, or the corner of a window in the image above. Unlike a random point on the background (sky) in the image above, the tip of the tower can be precise detected in most images of the same scene. It is geometrically (translation, rotation, ...) and photometrically (brightness, exposure, ...) invariant.

The extracted local features must be:

- *Repeatable* and *precise* so they can be extracted from different images showing the same object.
- *Distinctive* to the image, so images with different structure will not have them.

There could be hundreds or thousands of such features in an image. An image matcher algorithm could still work if some of the features are blocked by an object or badly deformed due to change in brightness or exposure. Many local feature algorithms are highly efficient and can be used in real-time applications. Due to these requirements, most local feature detectors extract corners and blobs.

Local Feature Detection and Description

There is a wealth of algorithms satisfying the above requirements for feature detection (finding interest points on an image) and description (generating a vector representation for them). They include

- Scale Invariant Feature Transform (SIFT)
- Speeded-Up Robust Features (SURF)
- Features from Accelerated Segment Test (FAST)
- Binary Robust Independent Elementary Features (BRIEF)
- Oriented FAST and Rotated BRIEF (ORB)

Task:

Download the set of images available in the zip folder of this assignment. There are three set of images available in the zip file at the following link. The link can be accessed using SEECS email address only. (https://drive.google.com/open?id=14jSNKLgHPBMmuV6VK0fofcrA97sYd10h)

- Set 1: book.jpg and book_person_holding.jpg
- Set 2: roma_1 and roma_2.jpg
- Set 3: building_1.jpg , building_2.jpg and building_3.jpg

Apply any of above local feature detector/descriptor for feature detection and find similarity between detected features. Show the similarity of top 10 matched features by drawing the line between the matched features.

Deliverable:

You may download PyCharm, an IDE to work for in Python. Implement your task using IDE

- The code files (.py files)
- A MS Word /PDF file containing following
  - Qualitative results i.e. a pair of images showing top 10/X number of matches by using the above mentioned local feature detectors/descriptors. You are suggested to use at least three of the feature detectors/descriptors.

There are three matching sets of images (building, roma and book). You need to demonstrate matching within sets only.

Task 2: Human Classification / Detection

Human Detection is a branch of Object Detection. Object Detection is the task of identifying the presence of predefined types of objects in an image. This task involves both identification of the presence of the objects and identification of the rectangular boundary surrounding each object (i.e. Object Localisation). An object detection system which can detect the class "Human" can work as a Human Detection System.

Human detection is an essential and significant task in any intelligent video surveillance system, as it provides the fundamental information for semantic understanding of the video footages. It has an obvious extension to automotive applications due to the potential for improving safety systems. Many car manufacturers (e.g. Volvo, Ford, GM, Nissan) offer this as an ADAS option in 2017. Some of the applications be given as;

- Self driving cars. Identifying pedestrians on a road scene
- Security. Restrict access for certain people to certain places
- Retail. Analysing visitors behaviour within a supermarket
- Fashion. Identify specific brands and persons who wear them

Histogram of Oriented Gradients

HOG is an acronym for Histogram of Oriented Gradients. It's an algorithm called a feature descriptor which helps with object detection in computer vision and image processing models. HOG is a kind of feature descriptor that counts occurrences of gradient orientation in localized portions of an image.

Task:

Download the INRIA Person Dataset Samples from here (https://drive.google.com/file/d/1pDr3138jwk8WuHF2CFGVIKbKvt27hMIH/view?usp=sharing).The link can be accessed using SEECS email address only.

The dataset contains positive and negative example of persons in the folders "Train" and "Test".

Compute HoG features and train two classifiers i.e SVM and Random Forest. You can use `sklearn` library's function `feature.hog` to compute HOG features. Experiment with different parameters and report the results obtained after training of both the classifiers i.e. Linear SVM and Random Forest Classifier. The classifiers are also available in `sklearn` library.

You may use sklearns' `joblib` package to save and reload your trained classifier model.

Deliverable:

You may download PyCharm, an IDE to work for in Python. Implement your task using IDE

- The code files (.py files)
- A MS Word /PDF file containing following
  - Quantitative performance measures i.e. F1 Score, Accuracy, True Positive Rate, False Positive Rates of both classifiers.
  - Qualitative results i.e. a few of the correctly classified images

There is a built-in feature descriptor+SVM classifer available with OpenCV. Please do not use it