CITS4402 Computer Vision

Project: Object Recognition in Cluttered Scenes

Due: Thursday, 29th May 2020, 4pm (NO EXTENSIONS)

Groups

You are supposed to have formed groups of 3 students per group and include the list of the members on LMS/Discussion Board as instructed on LMS/Announcements earlier.

Timeline

Friday 3rd of April 2020: Deadline for groups to be formed (see LMS/Discussion Board) and Project will be handed out.

Week 7 to Week 12 (see deadline below and on LMS to hand in your projects): Work on your project. You have 8 weeks (including 2 weeks of the teaching break) and your labs will be supervised as usual but **only** on **Tuesdays** and **Thursdays** sessions during the teaching weeks.

Week 12 during your regular lab session: Demo/presentation of the projects on zoom (10 minutes per presentation). Only one representative member for each group will present on behalf of the whole group. Please make sure your presentation is ready before you start.

Friday 29th May 2020 4:00 PM: Final deadline to submit your project (report and code) on LMS. This is the last day of the teaching semester so no extensions are possible.

THE PROJECT

You are required to develop an object recognition system for a robot. The idea is that a robot can get images or a video of its surrounding environment, but needs an intelligent computer vision software to process the images and find out what objects are present in the scene. In practice, the scene will have unwanted objects (clutter) and occlusions (objects of interest hidden behind other objects). Your system should account for these. A recognition system can be useful for example if you tell your robot to go to the classroom and bring your hat or your water bottle. The robot can also find its location and navigate based on known locations of fixed objects.

There are two phases to develop such a system **(1)** the learning phase (also called training phase) and **(2)** the testing phase.

Learning Phase/ Training Phase

1. Collect 20 objects that you want the robot to be able to recognize. We will call them objects of interest. Show a picture of these objects of interest in your report.
2. Take multiple images of each object in isolation. Take the images from different directions/orientations.
3. Extract features from each image of each object.
4. See if you can compress the features by projecting them into the PCA space.
5. Train a classifier using the original or compressed features.

Testing Phase

1. Put a few random objects of interest along with others that are not of interest to form a "scene".
2. Take an image of the scene. The scene should have clutter and some objects of interest should be partially occluded. You may consider three categories of scenes, Easy, Difficult, Very Difficult with 10 images in each category. In 'Easy' category there is no occlusion and less clutter (objects of interest are more than clutter), 'Difficult' category has some occlusion and equal number of clutter and objects of interest, 'Very Difficult' category has high Occlusion, more clutter and less objects of interest.
3. Extract similar features from the scene as above (as you did during the learning phase).
4. Compress the features if you did compression in the learning phase e.g. project them into the same PCA space.
5. Match all the features from the scene to the features of the objects of interest one by one. [You can do indexing here to speed up your search]
6. Based on the quality and consistency of the matches, your code should decide which objects of interest are present in the scene and find their locations. Display the outline of the object on top of the image scene (see demo from previous year below).
7. Repeat the above steps 30 times (i.e. generate 30 different scenes) and calculate the recognition rate of your system.

Develop a simple GUI to test your algorithm. The GUI should load a scene image one by one, find objects of interest in it and display the results. The recognition should be fully automatic and you may only specify the directory where the scene images are present. Your code should already know where to find the training data. Have a look at a demo from previous teaching years.

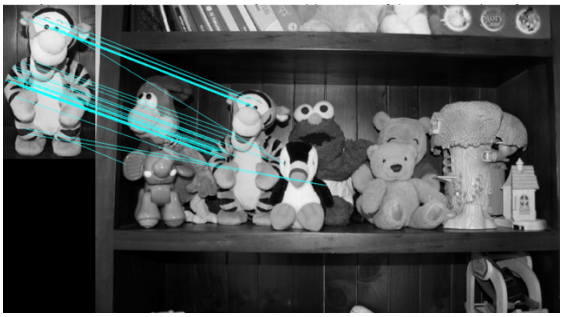
Note: You are **not** allowed to use the Matlab Computer Vision Systems Toolbox. The reason is that it is not available in your lab.

Tools and Tips

- You will be given SIFT (Scale Invariant Feature Transform) code to extract features but you are welcome to use other features (alone or in combination with SIFT).
- You are allowed to share your objects but not the images of the objects or the scenes.
- You may find the presentation helpful. You also need to watch the video lecture of week 7 entitled "Feature Detection & Extraction".
- Use the matching point pairs to find a transform between the two images.

Copyright: SIFT was developed by David Lowe. It is free only for educational purposes and commercial use requires a license.

Your recognition result for one object should look like the following:



Additionally, you should also display the outline of the object on top of the image scene as shown in the slides of the above presentation.

If multiple objects of interest are present in the scene, you should recognize them all. Calculate the final recognition accuracy as the number of objects found divided by the number of objects present in all scenes.

Distribution of Marks:

- Data collection: At lease 6 images each of 20 objects and 30 scene images (150 in total). [**5 marks**]
- GUI to test your code without any execution error. [**3 marks**]
- Feature extraction. [**3 marks**]
- Matching and display of correct matches. [**10 marks**]
- Localization of object (to be shown visually). [**4 marks**]

Presentations: [10 marks]

During your regular final lab sessions on **Tuesdays** and **Thursdays** (week 12), one representative of the group will present the system and demonstrate the recognition of objects.

Submission Requirements

You are required to upload (on LMS) a folder containing your report and code (in Matlab), training features, trained classifier and all the objects and scenes images. Marks will be based on your presentation, and the submitted data & code as per distribution of marks.