CSCM77 Computer Vision and Deep Learning

Coursework Assignment: Pac-Man

1. To be completed by students working individually.
2. Feedback: you will receive individual feedback at the end of your viva. An assessment and feedback form will be provided to you.
3. Learning outcome: The tasks in this assignment are based on your practical work in the lab sessions and understanding of the theories and methods. Thus, through this coursework, you are expected to demonstrate both practical skills and theoretical knowledge of several computer vision techniques.
4. Unfair practice: This work is to be attempted individually. You may get help from your lecturer, academic tutor and lab tutor, but you may not collaborate with your peers.
5. Submission deadline: The code must be submitted on Blackboard by **11am on the 30th of April**.
6. The work will then be marked by viva in the lab session **1pm Friday 10th May.**

The Task

In this coursework, you are given a set of 3D point clouds with appearance features (i.e. RGB values). These point clouds were collected using a Kinect system in our PhD lab. Several virtual objects are also positioned among those point clouds. Your task is to write a Python program that can automatically detect those objects from image and use them as anchors to navigate through the 3D scene. A set of example images that contain those virtual objects are provided. These example images are used to train a Random Forest based classifier in order detect the objects. Some Python code is provided to help you to complete the task. A demo code that shows how to obtain a 2D image by projecting 3D point clouds to the camera image plane after camera to re-positioning and rotation is provided as well.

The following materials from lectures are relevant to this task:

1. Camera translation and orientation
2. Feature descriptors, e.g. Histogram of pixel intensity, histogram of oriented gradients etc.
3. Supervised learning using Random Forest, CNN, Region Proposal Networks.

All the software and data are available on Blackboard.

Demo code:

Demo code is provided for orientating the camera view and obtaining an image from the current camera view. The software to generate and visualise the point clouds are also provided. Below, the Python functions that are required to train and test Random Forests are explained.

Python provides a suite of methods for Random Forest classification, namely the sklearn.ensemble.RandomForestClassifier class, which is able to train an ensemble for classification given some training observations and their labels. Key methods of the sklearn.ensemble.RandomForestClassifier class include:

- classifier = sklearn.ensemble.RandomForestClassifier(n_estimators=n_trees) # Instantiate a Random Forest object
- classifier.fit(data, labels) # Fit the Random Forest to the data and labels
- prediction_probability = classifier.predict_proba(data) # Get predicted class probabilities given data

The use of these functions will be further explained in the lab sessions.

Random Forests classification is required to detect those artificial objects among the point clouds. Training should be carried out on the provided example images. The detection requires a sliding-window based evaluation that is same as in human detection (HoG, covered in lectures). The sliding window should be the same size as the training images. You can use any features to train the classifier, e.g. histogram of pixel values or histogram of oriented gradients. Once a virtual object is detected, you need to move your camera to where the virtual object is located and start your search for the next one until all virtual objects are found. In the event that multiple virtual objects are detected in a single view, the nearest virtual object should be selected.

Viva:

To assess performance in this task you will give a live demo of your program with a member of the marking team. You will be asked key questions regarding certain aspects of the program's behaviour to ensure that you understand your presented solution. The viva must be carried out in the lab sessions before the deadline.

Marking Criteria:

You will be marked on several criteria for this task:

1. Extract images from the scene along the current direction of observation.
2. Generate informative features for the observation to allow detection of an object in view.
3. Ability to correctly train and use a classifier to detect the target in view.
4. Correctly moving to the position of virtual objects.
5. Correctly orientating the camera in the scene.
6. Number of virtual objects found.

An assessment and feedback form will be provided to you at the end of the viva. This form will be published on Blackboard as well.

Assessment: The percentages listed above indicate the approximate distributions of marks. **This assignment is worth 20%** of the total credit.