Introduction to Computer Vision (ECSE 415)
Assignment 2

Due: October 19^{th}, 11:59PM

Please submit your assignment solutions electronically via the myCourses assignment dropbox. Attempt all parts of this assignment. The assignment will be graded out of total of **75 points**. Students are expected to write their own code. (Academic integrity guidelines can be found at `https://www.mcgill.ca/students/srr/academicrights/integrity`). Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be graded.

1 Submission Instructions

1. Title two jupyter notebooks as (i) stitching (ii) classification.
2. Comment your code appropriately.
3. Do not forget to run Markdown cells.
4. Assume image folders are kept in a same directory as the codes. Make sure that the submitted code is running without error. Add a README file if required.
5. If external libraries were used in your code please specify its name and version in the README file.
6. Do not submit input/output images.
7. Answers to reasoning questions should be comprehensive but concise.
8. Submissions that do not follow the format will be penalized 10%.

Note that you can use any of the OpenCV functions shown during tutorial sessions for this assignment, unless stated otherwise.


(a)


(b)


(c)

Figure 1: Three different views of the same scene. (a) 1Hill.JPG (b) 2Hill.JPG (c) 3Hill.JPG

2 Image Stitching

2.1 Coding Questions

You are given three different views of the same scene in a folder 'stitching_images' (Figure 1). Follow these steps in order to stitch given images:

1. Compute **SIFT** keypoints and corresponding descriptors for images 1Hill and 2Hill. **(3 points)**
2. Find matching keypoints in two images and display the 20 best pairs. **(4 points)**
3. Find homography using RANSAC method and apply transformation to 1Hill. Image 2Hill should not be transformed. **(7 points)**
4. Stitch transformed 1Hill and original 2Hill together using **pyramid** image blending. **(8 points)**

Let's call the above resulting image 12Hill. Implement the following steps to stitch 12Hill and 3Hill.

1. Compute **SURF** keypoints and corresponding descriptors for images 12Hill and 3Hill. **(3 points)**
2. Find matching keypoints in two images and display the 20 best pairs. **(4 points)**
3. Find homography using RANSAC method and apply transformation to 3Hill. Image 12Hill should not be transformed. **(7 points)**
4. Stitch transformed 3Hill and 12Hill together and apply **linear** image blending. **(4 points)**

2.2 Reasoning Questions

1. What are the differences between SIFT and SURF? Comment on the speed, number of keypoints and density. Support your arguments with statistics. **(4 points)**

2. Discuss at least two pros and cons of pyramid and linear blending techniques. When should one prefer pyramid blending over linear blending? **(4 points)**

3. What would happen if a *random subset* of pairs of matching keypoints is used for finding homography? **(2 points)**

4. Discuss the relationship between the number of pairs of matching keypoints used for finding homography and the accuracy of inferred homography. **(2 points)**

3 Image Classification

Consider an automobile restoration company which receives thousands of vehicles per day. The company hires you to build a system which can automatically identify the manufacturer of the automobile so that it can be sent to the particular repair center. You recall the *Introduction to Computer Vision course* that you took during Fall of 2018 and decide to build a logo classification system using Histogram of Gradient (HoG) features.
The company provides you some images of different manufacturers' logos, as your training images that are taken in an ideal setting (i.e. no additional background or perspective distortion) and you are asked to test your built system on real-world images (i.e. with background and perspective distortion).

3.1 Coding Questions

For this task, you are given a set of training images (check 'dataset' folder): five logos of 'Ford' and five logos of 'Volkswagen' (see Figure 2). You are also given two real-world test images, one for each manufacturer. Use class label 1 for 'Ford' and -1 for 'Volkswagen'. Build a classification system using HoG features as follows:

- Training

    1. Resize the training images to 128×128. **(0.5 points)**
    2. Compute HoG features of size (32,32,8). Apply blocknorm in 4×4 cell neighborhood. (Deduce values of other required variables.) **(11 points)**
    (Suggestion: Make a function which takes list of images as arguments and delivers list of HoG features as output. The same function can be used during testing.)

Figure 2: Training Dataset

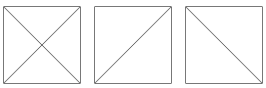

Figure 3: Sample images for question 3.2.2

3. Fit a nearest neighbor classifier with three neighbors. Use `KNeighborsClassifier` from `sklearn` library. **(2 points)**

- Testing

1. Resize test images to 128 × 128. **(0.5 points)**
2. Compute HoG features similar to what was done during training. **(2 points)**
3. Display the features for both test images. **(3 points)**
4. Classify the test images using the classifier built above. **(1 points)**

3.2 Reasoning Questions

1. Will HoG work if images of logos undergo random rotation? If yes, which property of HoG avails this feature? If not, discuss the sequence of computer vision technique(s) that can be used in order to regain uniform orientation of the logos. **(3 points)**

2. Consider that you are given a large dataset with three types of images: variable sized squares with either or both diagonals drawn (see Figure 3). How many and which directions of the gradient are *sufficient* in order to correctly classify the images into three categories? **(2 points)**