Fall 2019: CSI5139F
Assignment 1

Due: Friday, September 27th, 2019, 11:00pm in Virtual Campus
University of Ottawa - Université d'Ottawa

Jochen Lang

1 Image Labelling - Linear Classification

This assignment will give you a chance to familiarize yourself with Jupyter, scikit-learn and other ML tools. You will use the PHOS color image database of 15 scens under different illuminations. The database can be download from `http://robotics.pme.duth.gr/phos2.html` at the Laboratory of Robotics and Automation, Democritus University of Thrace. You are looking for an archive called `Phos2_0_2MP`. This archive contains downsampled images of 15 scenes under different illumination conditions and images with a diffuse illumination but taken with different exposure.

In this assignment, you will train binary and multiclass linear classifiers available from scikit-learn on this dataset.

1.1 Getting Started [3]

You will need to use the scikit-learn image processing tools to read and process the images. The images are stored in a directory structure with one directory per scene where the file name encodes the imaging condition.

- The scene is in the directory name, e.g., `Phos2_scene1`.
- The filename in the directory encodes the illumination condition, e.g., `Phos2_nonuni` or `Phos2_uni` for non-uniform illumination and diffuse or uniform illumination, respectively.
- The filename for the images with nonuniform illumination contains a simple count, e.g., `Phos2_nonuni_sc1_1` for scene 1 under non-uniform illumination 1.
- The filename for the images with different exposures specify the F-stop, e.g., `Phos2_uni_sc1_1_plus_1` means an aperture increase by one F-stop and hence the image appears slightly overexposed.

You will need to load all images and organize them suitably. For this you will need to load the images into numpy matrices. You will need to generate a list of filenames for the files to read. You can achieves this with hardcoding a loop for the filenames according to the above filename rules, or by reading the directory and file structure using the os package in python. Your jupyter notebook has to work with the unmodified directories and filenames.

The next step is reading and preprocessing the images. For this part, you will need to use the `skimage` package. Amongst others, the following commands will be helpful: `skimage.io.imread`, `skimage.color.rgb2gray`, and `skimage.rescale`. *Make sure to downsize the* `Phos2_0_2MP` *images further by rescaling them to* 1/5 *of their size.* For questions 1.2 and 1.3, please work with grayscale images.

1.2 Binary Classifier [3]

Train a logistic regression classifier to distinguish grayscale images with uniform vs. non-uniform illumination. Organize your data into training, validation and test set. State clearly in your Jupyter notebook what split you are using and give the number of images for each of the three sets. Also consider careful how to compose these sets, e.g., selecting images at random vs. some form of stratified sampling. Find and print the confusion matrix, the accuracy, the recall and the precision of your classifier based on the training data, as well as on the testing data. Print the ROC curve.

Annotate your Jupyter notebook with an explanation of the results. Without an explanation, you will not receive full marks.

1.3 Multiclass Classifier [2]

Train a logistic regression classifier but this time to classify all grayscale images into multiple (3) classes: underexposed ($-4, \ldots, -1$ EV), overexposed ($+1, \ldots, +4$ EV) and regular (all other images including non-uniform illumination). Re-organize your data into training, validation and test set. State clearly in your Jupyter notebook what split you are using and give the number of images for each of the three sets. Also consider careful how to compose these sets, e.g., selecting images at random vs. some form of stratified sampling. Find and print the confusion matrix and the accuracy. In your Jupyter notebook, discuss briefly how the classifier perform on this task, e.g., are there easier or more difficult classes, do certain classes tend to be confused etc.

1.4 Improved Classification of Colour Images [2 Bonus]

For the bonus, perform the multi-class regression of Section 1.3 but with colour images and try to improve the results further. Use an overexposed, an underexposed and a correctly exposed image not in the `Phos2_0_2MP` dataset and have it classified by your model from Section 1.3. Disuss the result. Make sure to discuss your improvements in the Jupyter notebook markup: No discussion, no bonus marks. Hint: Consider the classical computer vision pipeline.

2 Submission

You will need to submit your solution in a Jupyter file. Make sure you have run all the cells. All text must be embedded in the Jupyter file, I will not look at separately submitted text files. If your Jupyter file needs a local python file to run, please submit it as well. Assignment submission is only though Virtual Campus by the deadline. No late submissions are allowed, you can submit multiple times but only your last submission is kept and marked. Keep in mind that this is an individual assignment (no group work). Make sure to follow the rules of the University of Ottawa regarding academic integrity, plagarism and fraud.