Lab 10 - Object Detection and Colour Spaces

Exercises

Image Object Detection

17. As with the lab last week, run your python file using Python
18. Import OpenCV, numpy and matplotlib
19. Download a ".jpeg" image of your choice containing faces from Google images or elsewhere into the same folder as your python file. Load the image into OpenCV
20. As described in the link below, perform facial detection on the image of your choice. Verify that faces are being detected. You will have to download haarcascade_frontalface_default.xml and haarcascade_eye.xml from https://github.com/opencv/opencv/tree/master/data/haarcascades into your current directory. When downloading the XML from GitHub, make sure to download it in 'Raw' format.

    http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection

21. Using another of the trained xml classifiers described in https://github.com/opencv/opencv/tree/master/data/haarcascades, download another image of your choice and demonstrate the classifier operating on the image.

Colour channels:

22. Using one of your downloaded images, split the image into R,G and B channels and view the result using a matplotlib subplot (as described in the previous lecture)

Advanced exercises

3. Next, transform the image employed in step 6 into the HSV colour space. Once this has been performed, split the image into the separate HSV channels and view the resulting separated images. Documentation on how to transform from RGB to HSV can be found at:

http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor

4. Using the following tutorial (https://docs.opencv.org/3.4/da/d9d/tutorial_dnn_yolo.html), user the YOLO Deep Learning Network to classify an image of your choice.