

FIT5221 – Assignment 2 – Dr. Abhinav Dhall

Submission due – 7th August 2020 11.55PM Suzhou.

Instructions->

1. All code should be in Python 3.7.x and use *Skimage* imaging library only. OpenCV or PIL cannot be used.
2. There should one common *Report.doc* for all the tasks below.
3. Submit a single zip file containing the task python files, report and images.
4. Late submission penalty – 1 mark per day.
5. Submission is to be made only on Moodle.
6. Plagiarism cases will be dealt following Monash policy.

Task 1 (5 Marks total)

Your input image contains a single face. Convert the image into a cartoon. The face (and body) should be cartoonized. The background should be blurred. There can be various ways for cartoonizing like k-means based color quantization. However, the results may not be very convincing around the boundary. Another possible solution to use bilateral filtering (read it!).

The function name is *CartoonNizer_ID(image)* – 3 marks. It takes the input image as parameter and displays and saves the cartoonized output.

You may follow only **one of the following** three variants for completing the task above –

1. Your own Kmeans clustering based implementation – 1 mark
2. Bilateral filtering based implementation – 2 mark
3. Improved (your own/inspired) method and implementation – 3 mark

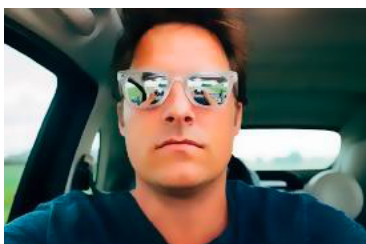
Bilateral filtering -

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html

Report containing details – 2 mark

Example →

INPUT



OUTPUT



Task 2 (5 Marks total)

The task is to find the number of rings (semi-circles, spheres and complete circles) in the attached image. Code function *CountShapes_ID(image)*. It takes the input image as a parameter and outputs the number of semi-circles, spheres and complete circles and also displays and saves the output image with highlighted objects.

CountShapes_ID(image) in python file– 3 marks

Report containing details – 2 marks

Image source - https://cdn.pixabay.com/photo/2017/01/06/15/46/ball-bearings-1958085_1280.jpg

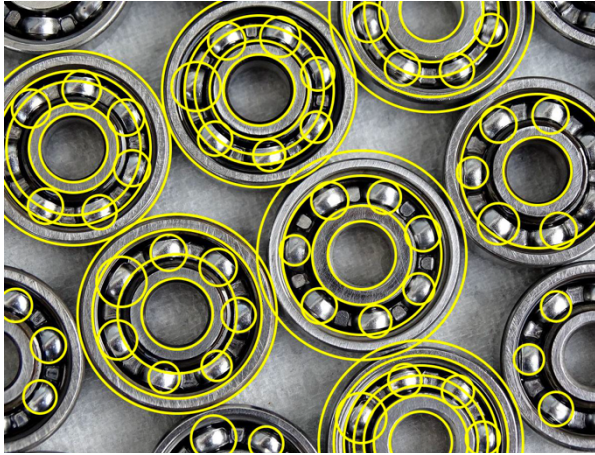
Hint – Read about Hough transform - <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>

EXAMPLE –

INPUT



OUTPUT



More assignment instructions ->

During the assignment marking the two task files will be executed directly and the outputs will be checked.

No change is going to be made to the code at our end. The image folder (for both the tasks above) should be part of the zip file, which you will upload on Moodle. In the image folder do not store more than 10 images. Make sure that the resolution of each image is $\leq 640 \times 480$.

Make sure that you check that the code works, when the zip file is unzipped.