

OpenCV Introduction

1) Setup openCV on ubuntu. Try to run the sample code provided. Compilation instruction is provided in readme.txt. The sample code should read the image as a command line argument and display its negative.

NOTE: If you have problems with installation we can settle it out in the double lab session or ask your friends who have done.

2) You need to edit the sample code and perform an intensity transformation on the input image and display both, the input image and the transformed image. Make intensity transformation as follows :

If $X < 20 \rightarrow 2 * X$

If $20 < X < 150 \rightarrow X/2 + 30$

If $X > 150 \rightarrow 2 * X - 195$

3) Modify the code and display the video sequence which is provided in "video_seq".

Image Interpolation

4) Input an image of size $M \times N$ and do the following

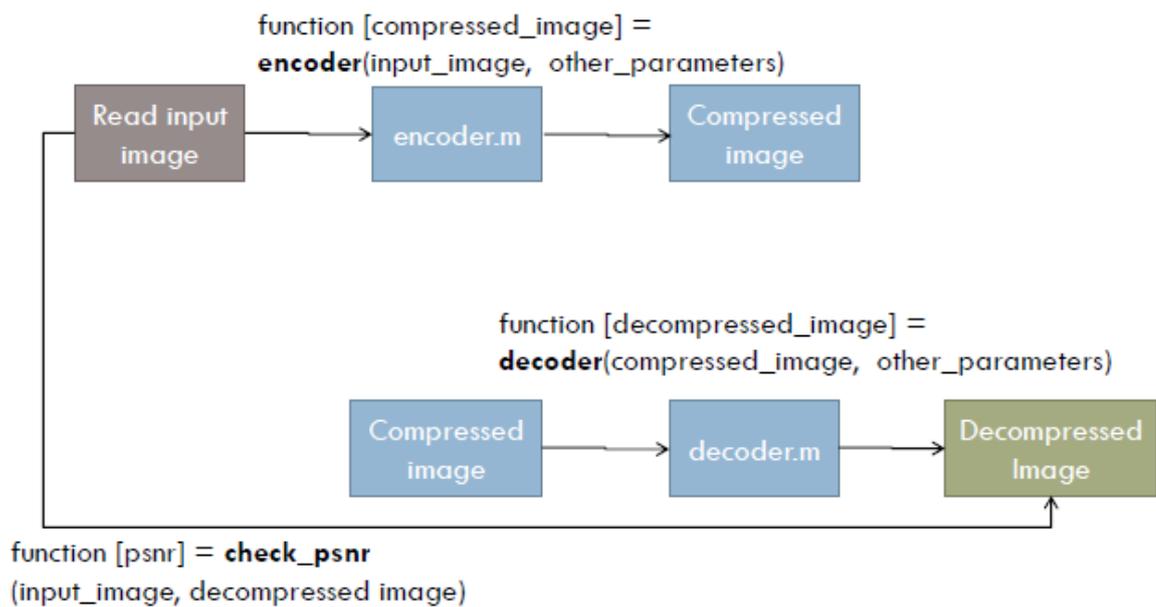
i) Scale factor of 0.5 and 0.2 using nearest neighbour.

ii) Scale factor of 2 and 5 using nearest neighbour.

Repeat i) and ii) using bi-linear interpolation and compute PSNR for each of the cases.

Use images : "bat", "woman", "gate". Also try with color image "lena".

Image Compression



5) Run Length Encoding is being used in PNG. You need to implement it as per the figure above (ie. Encoder and decoder). Take an input gray image as an $M \times N$ array. This would mean that, uncompressed image takes $M \times N$ bytes. Compress it with RLE, you would create 2 arrays one for run-count and another for symbol. And suppose these resulting arrays are of size P , would mean that you have achieved a compression of $(2 \times P) / (M \times N)$. Size of arrays to be computed with `size()`. Compute PSNR values. State all observations and tabulate the results.

Image	Compression Ratio	MSE	PSNR

6) Do similar activity for bit-plane-encoding. Try removing 1st, 2nd, 3rd least significant bit planes.