

Homework #1  
Due Tu. 2/04

You must turn in your code as well as output files. Please generate a report that contains the code and output in a single readable format.

1. Getting Started

- (a) Download the “standard” test images from the Gonzalez and Woods website.  
[http://www.imageprocessingplace.com/root\\_files\\_V3/image\\_databases.htm](http://www.imageprocessingplace.com/root_files_V3/image_databases.htm)
- (b) Download the sample images from the class website.  
<http://www.ee.unlv.edu/~b1morris/ecg782/hw/hw01>

2. Histogram Equalization

- (a) Write a function `hist_eq.m` that performs histogram equalization on an intensity image. The function should take as inputs an intensity image and the number of gray level value bins. Create a separate m-file for this function.
- (b) Perform histogram equalization on the jetplane image using 256, 128, and 64 bins. Compare the original image and the histogram equalized images by plotting the corresponding histograms and images side-by-side in a  $2 \times 2$  subplot matrix.
- (c) Perform the equalization in  $32 \times 32$  blocks. Display the output image. You will find `blockproc.m` useful.

3. Basic Morphology

- (a) Threshold the image `SJEarthquakesteampic.jpg` to detect faces. Be sure to describe how you obtained your threshold. You may find this is easier in another colorspace such as HSV.
- (b) Use morphological operations to clean the image. Count the number of players in the cleaned threshold image.
- (c) Create an output image that has a bounding box around each face. Use `regionprops.m`.

Extra Repeat for `barcelona-team.jpg`.

4. Filtering

- (a) Consider image `DSCN0479-001.JPG` as a perfect image. Add white Gaussian noise with variance 0.005. Smooth with a  $3 \times 3$  and  $7 \times 7$  box filter and a median filter. Compute the mean squared error (MSE)

$$MSE = \frac{1}{MN} \sum_m \sum_n (I_1(m, n) - I_2(m, n))^2$$

and the peak signal-to-noise ratio (PSNR)

$$PSNR = 20 \times \log_{10}(255/\sqrt{MSE})$$

for the noise reduced images. Which filter has the best results based on the error measures? How do the results compare visually?

- (b) Repeat (a) with salt and pepper noise with noise density 0.05.
- (c) Do the filtering again but this time on a real noisy image `DSCN0482-001.JPG` obtained at higher ISO. Compare the results visually only this time. Which filter works best for “real” noise? How much time does each type of filter require (use `tick.m` and `toc.m`)?