Homework 2

(Total 230 pts)

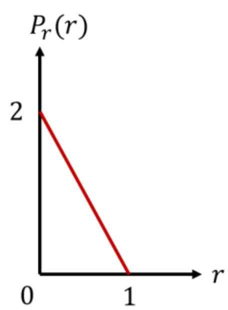
Due 5:00 pm, October 11, 2024 (Friday)

Canvas submission as a single PDF file ('hw2_lastname.pdf')

1. (10 pts) A random variable X has a probability density function (PDF) given by
$$f_X(x) = \begin{cases} 1, & 1 < x < 2 \\ 0, & \text{elsewhere.} \end{cases}$$
Another random variable Y is related to X by $Y = X^2$. Determine the numerical value of the mean square value of Y, $E[Y^2] = \int_{-\infty}^{\infty} y^2 f_Y(y) dy$. Round your answer to the first place to the right of the decimal point.

2. (30 pts) In Matlab, use the *rand* function to generate 1,000,000 samples of the random variable X with uniform distribution in the interval (1, 2). Let $Y = X^2$.
Hint: you can shift the values of the random samples directly generated by *rand* to the desired interval by adding an offset of 1.
(a) Show the empirical distribution of X by using the *histogram* function (with 'pdf' normalization).
(b) Show the empirical distribution of Y by using the *histogram* function (with 'pdf' normalization).
(c) Estimate the mean square value of Y by using the *mean* function.
Attach your Matlab script, the histograms, as well as a screenshot showing the result of running your script to obtain the estimate of $E[Y^2]$.

3. (30 pts) An image with continuous intensities in the range [0,1] has the PDF $p_r(r)$ shown below.



(a) Determine the transform $s = T(r)$ such that $p_s(s) = 1$.
(b) Plot the waveform $T(r)$.
(c) Show that by using the transform obtained in (a), we can indeed achieve $p_s(s) = 1$.

4. (40 pts) Suppose a 2-bit image ($L = 4$) of size 10×10 has the intensity distribution shown in the table below.

r_k	n_k
$r_0 = 0$	40
$r_1 = 1$	30
$r_2 = 2$	20
$r_3 = 3$	10

(a) Plot the intensity distribution $p_r(r_k)$ of this image.
(b) Determine the histogram equalization transform $s_k = T(r_k)$.
(c) Plot the transform function $s_k = T(r_k)$ by using the "stairs" function in Matlab.
(d) Plot the equalized histogram $p_s(s_k)$.
(e) Attach your Matlab script.

5. (30 pts) Load the built-in image "pout.tif" to Matlab.
   (a) Display the image.
   (b) Plot the full histogram by using
   ```
   >> stem(bins,counts)
   ```
   where bins and counts are obtained from running "imhist".
   (c) Calculate the standard deviation of the image.
   (d) Equalize the histogram of the image using "histeq". Display the resulting image J.
   (e) Plot the full histogram of the above image J.
   (f) Calculate the standard deviation of the above image J, and attach the Matlab script.

6. (60 pts) Image De-noising. In astronomy, imaging under very low light levels often cause sensor noise to render individual image virtually useless for analysis. We will simulate the generation of Gaussian white noise corrupted images in Matlab and use the method of image averaging to reduce the noise.
   Download the Galaxy Pair NGC 3314 image from:
   https://apod.nasa.gov/apod/image/0505/ngc3314_keel_big.jpg
   (a) Convert the image to a grayscale image by using the function *rgb2gray*. Display the resulting grayscale image J.
   (b) Generate a noise corrupted image by
   J_noisy = double(J) + std*randn(size(J));
   where the *std* (standard deviation) of the Gaussian noise is 64 pixel intensity levels.
   (c) Display the noisy image by using: imshow (J_noisy). Attach the image.
   (d) Display the noisy image by using: imshow (J_noisy, [ ]). Attach the image.
   (e) Comment on why the same image were displayed differently in (c) and (d).
   (f) Generate N = 10 noisy images in the same way as in step (b). Calculate the average image J_avg. Note: J_avg should be of 'double' type. Do NOT truncate J_avg by conducting double to uint8 conversion.
   Display this image by using: imshow (J_avg, [ ]). Attach the image;
   (g) Calculate the difference image by using: diff = J_avg - double(J);
   Then calculate the variance of the difference image.
   (h) Repeat (f) and (g), with N = 100 noisy images.
   (i) Repeat (f) and (g), with N = 1000 noisy images.
   (j) Fill in the table below with the above variances of difference images, as well as the ratios (noise variance/N). Also, comment on how the noise reduction performance changes with the number of noisy images for averaging). Attach your Matlab scripts.

N	10	100	1000
Variance of diff			
$\frac{Noise\ Variance}{N}$			

7. (30 pts) A truecolor image, also known as an RGB image, is an image in which each pixel is specified by three values -- one each for the red, blue, and green components of the pixel's color. In the following, we construct a single RGB image using three representative bands from a hyperspectral image.

   Read the hyperspectral image data 'paviaU.dat' into Matlab using the *hypercube* function, and use the following three representative bands to represent the color components.

Band Index	**Color Component**
56	R
31	G
11	B

   For each color component, first normalize the pixel values into the dynamic range [0, 255], and then convert the data type from ‘double’ to ‘uint8’. More specifically, each color component image f will be normalized into its scaled version I, where $f_m = f - \min(f)$, and $I = \mathrm{uint8}\left(255 \times \frac{f_m}{\max(f_m)}\right)$.

   Next, combine the scaled versions of the three color components and display the resulting RGB image. Attach your Matlab script and the image displayed.