Problem 1 (15 points): Gray level modification: Create an ImageJ plugin GrayLevel_Modification that modifies the values of an 8-bit grayscale input image according to the function \( s = 16 \sqrt{r} \), where \( r \) is the input intensity and \( s \) is the processed intensity. The factor of 16 guarantees that the result will be in the range 0 to 255. Use this plugin to modify any currently open 8-bit grayscale image.

Problem 2 (15 points): (Thresholding): Create a plugin called Median_Threshold that sets the threshold value to the median of the histogram. Use this plugin to threshold an input image 8-bit grayscale image. Recall that the median \( m \) is the value that satisfies \( P(x < m) = P(x > m) \), in other words, half of the intensity values are less and half of the values are greater than \( m \).

Problem 3 (15 points): Power Transform: Create an ImageJ plugin Power_Transform which performs a power law transformation on an image. This function should an 8-bit grayscale image and the gamma value (as a variable you can modify in your program) to transform the image. Remember that power law transformations are achieved using the simple formula:

\[
s = c \times r^\gamma
\]

where \( s \) is the processed pixel value, \( r \) is the original pixel value, \( \gamma \) is the parameter controlling the power law transformation and \( c \) is a constant usually set to 1. Try this new function out on the following images (spine.jpg and runway.jpg), experimenting with different values for \( \gamma \). In the comments of your plugin, state what values of \( \gamma \) worked best for each spine.jpg and for runway.jpg.

Problem 4 (25 points): Burger and Burge problem number 4.3 (page 51). For this problem, you should create a plugin called NonLinear_Binning that solves the problem described. Create a table of 10 arbitrary ranges and pick appropriate intervals. It should be possible for the ranges you choose in your program to be changed and the program recompiled.

Problem 5 (30 points): Burger and Burge problem number 5.1 (page 83). For this problem, you should create a plugin called AutoContrast_quantiles.

Submitting Your Work

Submit all ImageJ plugins (GrayLevel_Modification.java, MedianThreshold.java, Power_Transform.java, NonLinear_Binning.javam and AutoContrast_quantiles.java) put into ONE zip file named yourfirstname_yourlastname_hw1.zip using turnin. DON'T EMAIL ME YOUR HOMEWORK. Also, TEST YOUR CODE IN THE ZOOLAB BEFORE SUBMITTING.