UNIVERSITY OF
LINCOLN

Lincoln School of Computer Science

Assessment Component Briefing Document

Title: CMP3108M Image Processing, Assessment Item One – Image Processing	**Indicative Weighting: 50%**

Learning Outcomes:

On successful completion of this component the student will have demonstrated competence in the following areas:

- [LO1] critique the theoretical knowledge of image processing, including how to process and extract quantifiable information from images.
- [LO2] apply image processing techniques to solve practical problems.

Requirements

This assessment comprises two assessed components, as detailed in the following page.

1. A report (in PDF format) that describes your approach to the tasks (maximum 12 pages, including figures but not the cover page). Weighting: 50% of this assessment.
2. A file containing all functions written in the MATLAB code with clear comments and requested figures. Weighting: 50% of this assessment.

Useful Information

This assessment is an individually assessed component. Your work must be presented according to the Lincoln School of Computer Science guidelines for the presentation of assessed written work. Please make sure you have a clear understanding of the grading principles for this component as detailed in the accompanying Criterion Reference Grid.

If you are unsure about any aspect of this assessment component, please seek the advice of a member of the delivery team.

Submission Instructions

The deadline for submission of this work is included in the School Submission dates on Blackboard.

You must make an electronic submission of your report in PDF format together with a zip file containing all source code files by using the assessment link on Blackboard for this component. The report should be submitted through TurnItIn and the zip file should be uploaded as supporting material. You must attend the lectures for further details, guidance and clarifications regarding these instructions.

DO NOT include this briefing document with your submission.

Task: Image Processing Assignment

Download and unzip the file ‘Assignment_Input.zip’ from Blackboard. You should obtain:

- Four input images
- Four MATLAB script m-files named ‘Task1.m’ to ‘Task4.m’.

Complete the MATLAB m-files to perform the corresponding tasks described below. As a guide, a few command lines for performing Task1 have already been added to the script. You need to add the command lines to implement the other steps. Please add appropriate comments to your code to briefly explain what each section is doing.

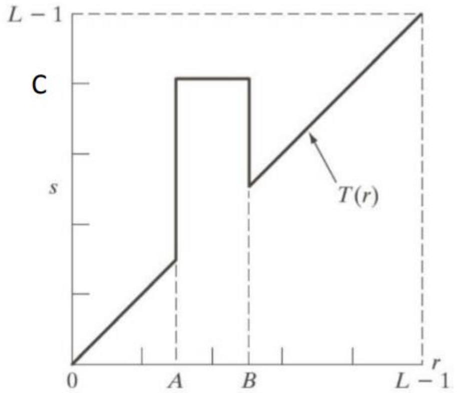
You must make an electronic submission of your report in PDF format. The electronic submission should also include all four MATLAB m-files, as supporting materials, which produce the desired results and display the outputs. Make sure the MATLAB scripts do not throw any error or display any error message. Put all the files (including the input images in a folder, and compress the folder into a zip file for submission. Name your zip file and PDF report using this format: LastName_ FirstName_StudentNo (for example: Zolgharni_Massoud_12345678).

Task1 – Interpolation (20 pts)

Complete the MATLAB script to load the image 'Zebra.jpg' and convert it to grey-scale. Then resize the image from its original size of 556×612 to an enlarged size of 1668×1836 by interpolation. Implement both nearest neighbour and bilinear interpolation. Display both re-sized images in your report. Also add at least one close-up (zoomed-in section) to you report where the difference between the two interpolation techniques is clear. Discuss the differences you notice between the two techniques. For this task, you CANNOT use the MATLAB built-in functions ‘*imresize*’ and ‘*interp2*’. However, you CAN use any other built-in function, if necessary.

Task2 – Point Processing (15 pts)

Complete the MATLAB script to load the image ‘SC.png’ and apply the following Piecewise-Linear transformation function to the image. Assume the diagram is drawn according to scale. This transformation highlights range [A, B], but preserves all other grey levels (identity). You can use the following values: A=80, B=100, and C=220. For this task, you CAN use any MATLAB built-in function. Add figures of the original and transformed images to your report.



Task3 – Neighbourhood Processing (20 pts)

Complete the MATLAB script to load the image ‘Noisy.png' and convert it to grey-scale. Then implement smoothing filters using averaging and median filters with a kernel (mask) size of 5 (neighbourhood of 5×5). Use zero-padding to deal with pixels on the edges of the image. For this task, you CANNOT use the MATLAB built-in functions ‘*fspecial*’, ‘*imfilter*’, ‘*conv2*’, ‘*medfilt2*’ and ‘*filter2*’. However, you CAN use any other built-in function, if necessary. Discuss the differences you notice between the effects of the application of two filters on the image. Display both filtered images in your report.

Task4 – Object Recognition (45 pts)

Complete the MATLAB script to load the image ‘Starfish.jpg’ and, through a series of image processing techniques you choose, generate a binary image where zero means no starfish detected and a non-zero value means that the pixel belongs to a starfish as shown in the figure below. For this task, you CAN use any built-in function. (Tip: you can separate the starfishes by the area and roundness of the recognized objects. First estimate each object's area and perimeter. Then use a simple metric indicating the roundness of an object: metric = 4*pi*area/perimeter^2).

In your report, explain each step you have taken and fully justify the methods you have used. Illustrate the outcome of each processing stage by adding figure(s) to your report. For example, if you used a particular type of spatial filtering, you need to explain why you have chosen it and discuss its effect by showing the figures of the original and filtered images. Add an explanation/interpretation of your results.