MOD002643 Image Processing Coursework 2018/9:

A System for the Recognition of Hand Gestures (Deadline Friday 21st December, 2018)

Section A (60 Marks)**:**

Example outputs for Section A are provided in Appendix A. For each question in Section A, provide a source code script (properly formatted, see Appendix B), any numerical measurements requested, and a brief evaluation of the success of the operation. In total, the text for section A should not exceed 500 words. Note: these tasks do not require loops if implemented efficiently.

A1. First we will prepare a test image of your hand. You might like to remove rings, wristwatches, and other coloured objects that may interfere with detection. Take one set of coloured fingertips (supplied), and keep them safe. If you need a replacement set, you will need to purchase them.

Take a colour photograph of your left hand, palm flat down against a sheet of white A4 paper on which you have written your SID number written in black ink, attempting to capture only the paper (rather than the surrounding table), like the example shown in Fig. 1. A template sheet (for printing) with a box in which to write your SID number is provided in the file **`template.pdf`**.

Note the order with which the coloured fingertips are to be placed on the fingers. The fingertips (properly called the *distal phalanx*) should be worn in the following order: orange for *pollex* (thumb), yellow for *digitus secundus manus* (index finger), blue for *digitus me'dius* (middle finger), green for *digitus annula'ris* (ring finger), and red for digitus mi'namus ma'nus (little finger).

Once you have your image, load it into MATLAB, produce a greyscale version, and then display both images side by side (see Fig. 1). State the equation and name of the greyscale conversion.

(10 marks)

A2. Next we collect and display some statistics relating to the images from question A1. We will plot normalised histograms of the RGB and greyscale images, and report the extrema (max and min) and arithmetic mean for each colour channel (R, G and B) and the greyscale image. Examine your histogram and collect a new test image if it seems to be under- or overexposed. Note that you can plot lines for the red green and blue channels on the same axis for the RGB histogram if you use lines rather than bars (see Fig. 2). Axes should be fully labelled and appropriately scaled. Plot the arithmetic mean of each colour channel, and the greyscale mean, as vertical dotted bars on the respective histograms.

(10 marks)

A3. Next, use high, low, band pass, and band reject thresholding on the RGB channels of the colour image (as required), to produce five binary images (thumb, index, middle, ring and little) that isolate, as well as possible, the five coloured fingertips into five separate images wherein 1 = in fingertip and 0 = not in fingertip. Present these in a figure, along with isolated RGB pixels from the original colour image (see Fig. 3, which shows an imperfect outcome). Report the thresholds and type of thresholding used. Note that you will need to experiment to find red, green and blue threshold values and threshold types to achieve the desired effect.

(10 marks)

A4. Next, implement code that places the coloured fingertips on top of the greyscale version of the hand image, to emphasise their locations (see Fig. 4). Also, in this question, report the size of the five fingertip pixels clusters identified (i.e., the cardinality of each set)

(10 marks)

A5. Next, calculate the average coordinate of each of the five pixel clusters, and display the image created in A4 annotating with 5 circles (for the five average pixel cluster coordinates), and four green lines connecting adjacent fingers (see Fig. 5). Report the coordinates of the five cluster averages and the lengths of the 5 lines (i.e., the Euclidean distance between each adjacent pair of fingertips). To accomplish this, you may need to investigate the **`find`** function.

(10 marks)

A6. Next, using the RGB image from question A1, devise thresholds that isolate the skin from the background. You may wish to conduct some research to determine appropriate values to use, and should consider constructing versions of the colour image in alternative colourspaces, such as HSV (hue, saturation and value). Once you have achieved satisfactory segmentation (i.e., a binary image in which the skin is 1 and the background and fingertips are 0), apply the following edge detection methods to see which yields the most accurate edge contour along the outside boundary of the hand: 1. Robert's, 3. Prewitt, 4. Sobel, 4. Canny. Take the contours produced by these methods and trace a red line around the original RGB image (see Fig. 6, which shows an imperfect outcome).

(10 marks)

Section B (40 Marks)**:**

For each question in Section B, provide code, test output figures, and text describing your approach and how well the solution operated. In total, the text for Section B should not exceed 1000 words (e.g., 250 words per question).

B1. Collect another set of images with your left hand palm *upwards* this time in which you make the following five hand gestures: (a) flat hand, (b) thumb touching index finger curved down into palm, (c) thumb touching middle finger curved down into palm, (d) thumb touching ring finger curved down into palm, (e) thumb touching little finger curved down into palm. Using the code you developed above, and some additional logic (e.g., based on Euclidean distance between fingertip clusters), be able to determine which one of these five gestures is represented in the image loaded.

(10 marks)

B2. Evaluate whether the use of morphological operators improves the identification of the binary clusters belonging to each fingertip (i.e., remove false positives and closes small holes) from Section A, and whether this improves the estimation of the central coordinate of each. Describe the approach taken and provide test output images. Next, evaluate whether morphological operators improve the classification of the gestures described in question B1.

(10 marks)

B3. Evaluate whether the use of hysteresis thresholding improves the identification of the binary clusters belonging to each fingertip (i.e., remove false positives and closes small holes) from Section A, and whether this improves the estimation of the central coordinate of each. Describe the approach taken and provide test output images. Next, evaluate whether hysteresis thresholding improve the classification of the gestures described in question B1.

(10 marks)

B4. Collect test images and develop code for classifying a range of additional gestures based upon the relative positions of the five fingertips – for example, can gestures corresponding to the 10 decimal digits (0..9) be interpreted? Additional marks will be awarded where this code operates in real-time using a webcam input, perhaps with a mouse click or button press to indicate that the gesture is ready for interpretation. You may wish to start by looking at the `snapshot` function, which acquires single frames from the computer's webcam.

(10 marks)

Your report must be printed and submitted to the iCentre by the published deadline.

Electronic only submissions will not be accepted.

Since your report will contain colour images, you will need to use colour printing.

You must print your code (for each question), not only the written evaluation and test results. This is so that comments can be written on it during marking.

Appendix A: Example Outputs for Questions A1..A6

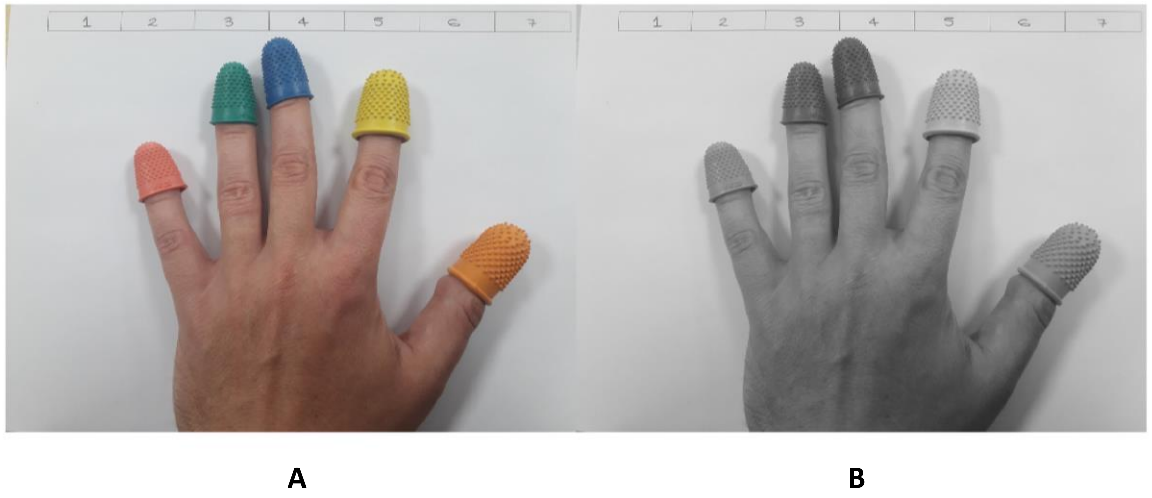

Figure 1: Example Output from Question A1 (Test Image in **A**. RGB and **B**. Greyscale).

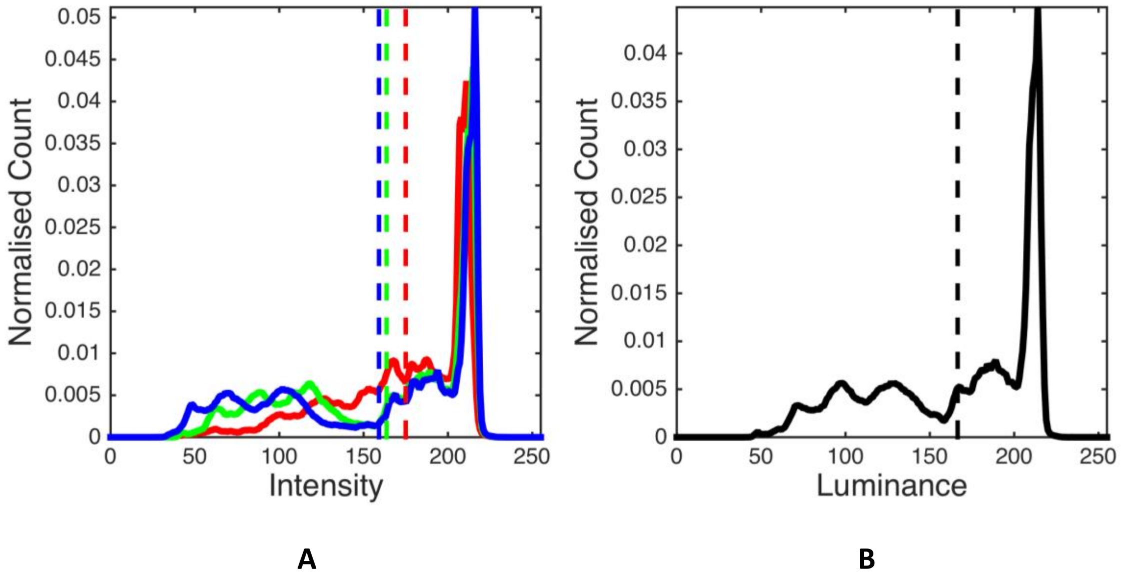

Figure 2: Example Output from Question A2 (**A**. RGB and **B**. Greyscale Normalised Histograms).

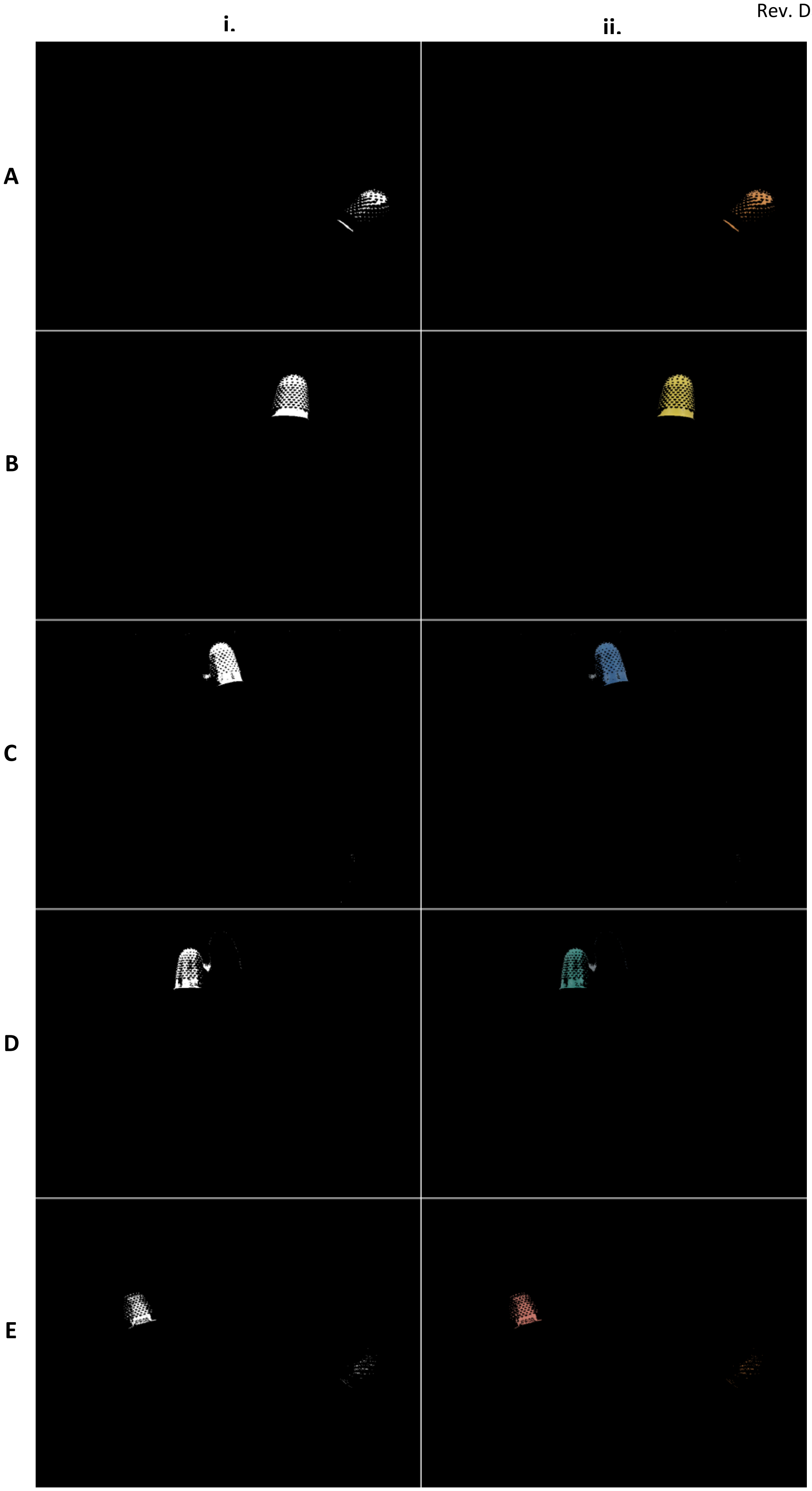

Figure 3: Example Output from Question A3 (Thresholding to Isolate Fingertips **A**. Thumb, **B**. Index Finger, **C**. Middle Finger, **D**. Ring Finger, **E**. Little Finger, where **i**. binary masks, **ii**. isolated pixels).

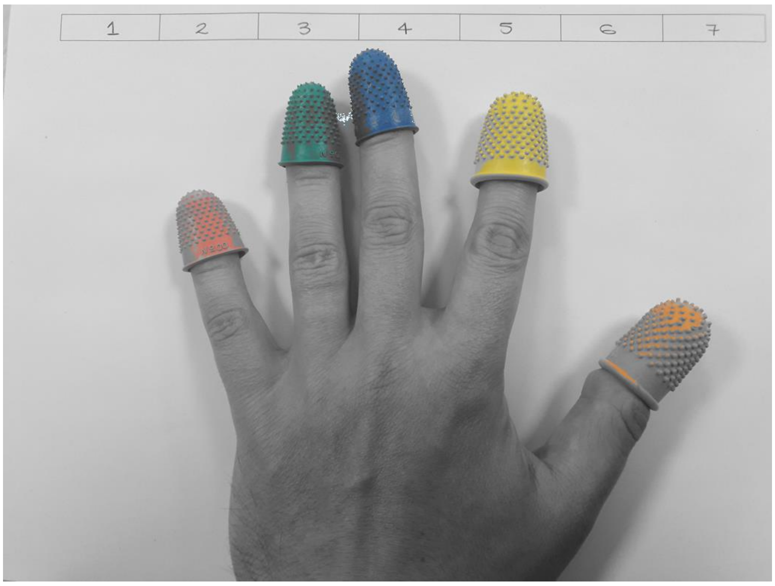

Figure 4: Example Output from Question A4 (Isolated Coloured Fingertip Pixels Superimposed on Greyscale Image).

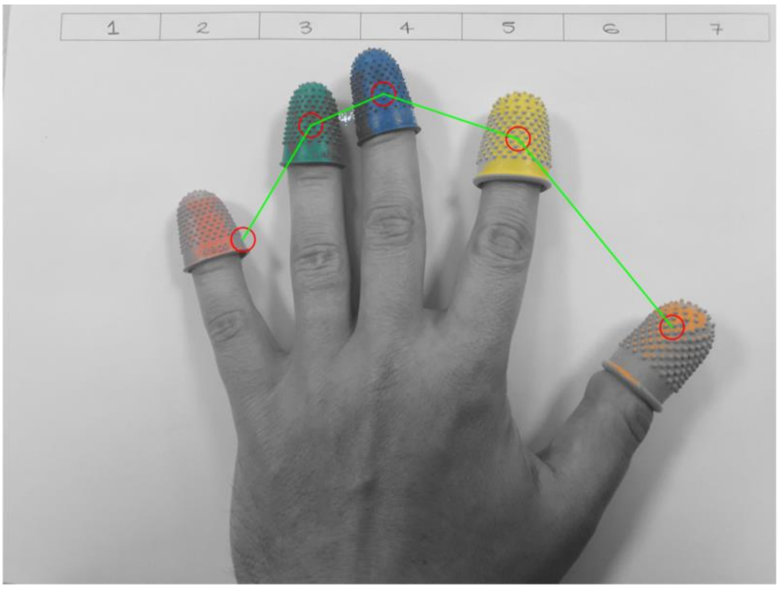

Figure 5: Example Output from Question A5 (Identify Centre of Pixel Clusters, Annotate, and Join Centres with Euclidean-distance Lines).

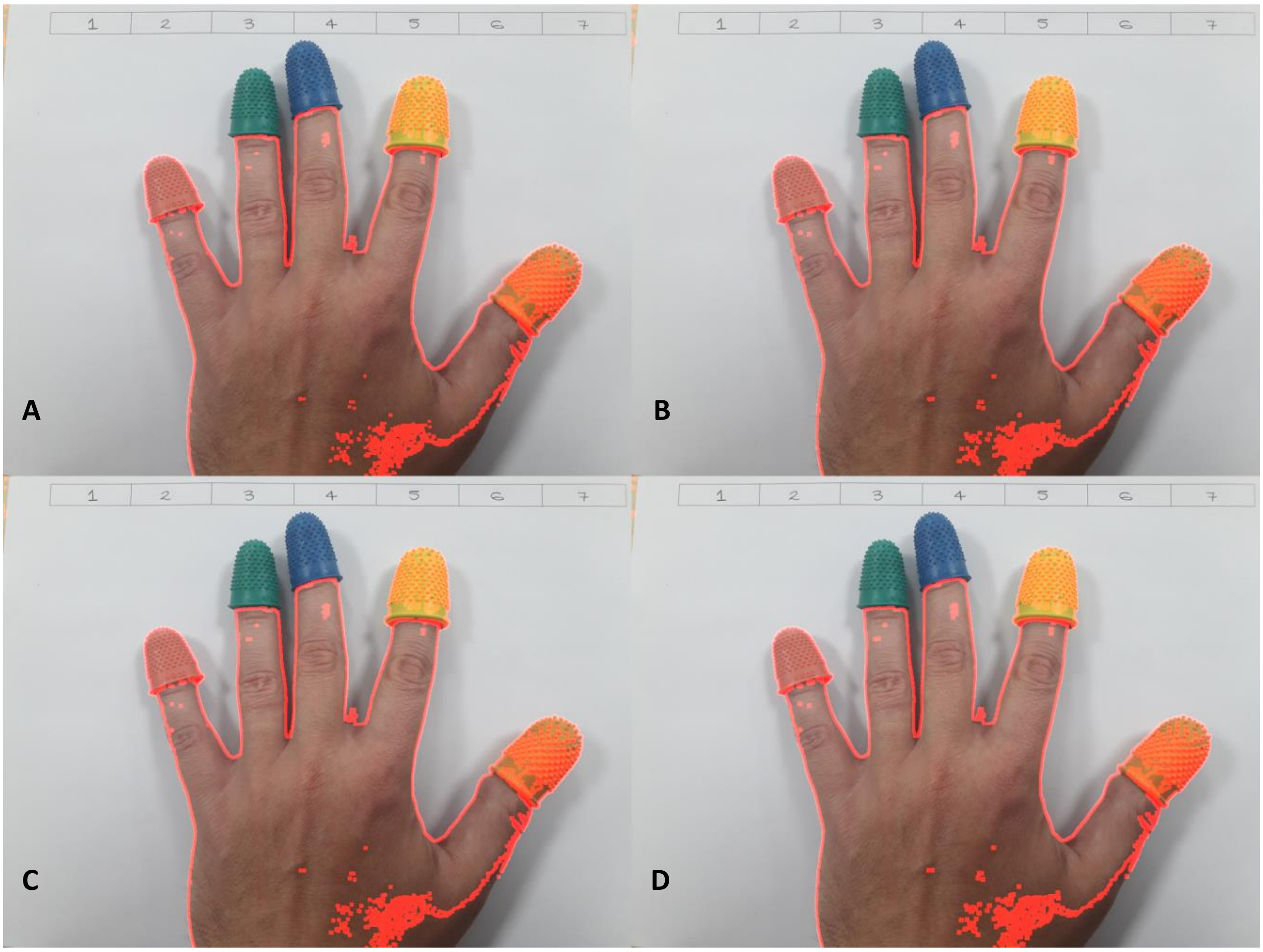

Figure 6: Example Output from Question A6 (Trace Boundary of Skin using **A**. Robert's, **B**. Prewitt, **C**. Sobel, and **D**. Canny operators, and Annotate RGB Image with Red Edge Contour).

Appendix B: Reporting Standards

Standards for Report Presentation:

1. Any equations should be properly typeset in Word using the equation editor.
2. Figures presented in your report should be properly exported from MATLAB in high resolution using the print command (e.g., `print –f1 –r300 –dbmp Q1.bmp`), or saved using `imwrite` function, and then inserted into your Word document (i.e., no screenshots please).
3. Figures, graphs and charts should be properly annotated and labelled (e.g., x and y axes in graphs should always have labels and the units should be stated, where appropriate).
4. Stick to the word limit prescribed (500 words per section for section A, not including figures, tables, equations, references or code, and 1000 words for section B). Write in the third person, and be concise and accurate.
5. Figures and tables are to be numbered sequentially and have an explanatory caption. Equations should be referred to in the text (see Tutorial 01 for an example of how this is done).
6. Your report must be printed and submitted to the iCentre by the published deadline.

Standards for Code Presentation:

1. Variable names should make sense (i.e., avoid single letter variable names, except where these correspond to mathematical convention, like M, N, i and j, or where variable names correspond exactly to the variables used in equations that you provide). Variable names should not be verbs. Where it is appropriate for variables to contain several words, use camelCaps (first letter lower case, first letter of each subsequent word in upper case).
2. Code should be well commented (in MATLAB, a comment starts with the % symbol). Where the end keyword is used to terminate a loop or if statement, place % end if, % end while, or % end for, as appropriate, to highlight what the end corresponds to.
3. Break large programs into functions, where appropriate, such that each function is informatively named, starting with a verb, and performs one well-defined task.
4. Avoid using literals in programs; either use arguments to control settings (e.g., inputs to functions), or collect settings from the user at run-time, or declare named constants (by convention, in upper case).
5. Code should be properly indented, and ordinarily one line should contain only one command. Before submitting your work, highlight all code and click “smart indent”.
6. Ensure code is concise, efficient, and that unnecessary work is not done. Check that your program does not perform calculations or declare variables that are not subsequently used.
7. It is acceptable to use built-in functions provided in MATLAB or its official toolboxes, if these are adequately explained in the corresponding text. If you use MATLAB code from a third-party source, this source should be acknowledged, and the underpinning theory precisely explained.
8. Code should be printed and included in the report you submit to the iCentre. Also include a USB device or CD/DVD containing your code and test image(s).

Appendix C: Mark Scheme

Section A

Task	Code Functionality	Presentation of Results	Written Evaluation	Adherence to Standards	Total
A1	0..4	0..2	0..2	0..2	0..10
Comments					
A2	0..4	0..2	0..2	0..2	0..10
Comments					
A3	0..4	0..2	0..2	0..2	0..10
Comments					
A4	0..4	0..2	0..2	0..2	0..10
Comments					
A5	0..4	0..2	0..2	0..2	0..10
Comments					
A6	0..4	0..2	0..2	0..2	0..10
Comments					
TOTAL (0..60)					

Section B

Task	Code Functionality	Presentation of Results	Written Evaluation	Adherence to Standards	Total
B1	0..4	0..2	0..2	0..2	0..10
Comments					
B2	0..4	0..2	0..2	0..2	0..10
Comments					
B3	0..4	0..2	0..2	0..2	0..10
Comments					
B4	0..4	0..2	0..2	0..2	0..10
Comments					
TOTAL (0..40)					