



UNIVERSITY OF SOUTH AUSTRALIA

Assignment Cover Sheet – Group

STEM Academic Unit

An Assignment cover sheet needs to be included with each assignment. Please complete all details clearly.

Student group Name: Group 5

Student Name: (write as per your ID card)		
1.	Samuel MALONEY	110 283 755
2.	Lining HOU	110 253 108
3.	Kane O'Brien	110 268 293
Course code and title:		MFET 4015 (Machine Learning and Vision Systems)
Assignment title		Group Assignment 2022
Course Coordinator:		Associate Professor Sang-Heon LEE
Submission due date:		12/ 6/ 2022
Academic Program: (circle applicable)		Undergraduate:
		Postgraduate:

We declare that the work contained in this assignment is our own, except where acknowledgement of sources is made. We authorise the University to test any work submitted by us, using text comparison software, for instances of plagiarism. We understand this will involve the University or its contractor copying our work and storing it on a database to be used in future to test work submitted by others.

We understand that we can obtain further information on this matter at
<http://www.unisanet.unisa.edu.au/learningconnection/student/studying/integrity.asp>

Note: The attachment of this statement on any electronically submitted assignments will be deemed to have the same authority as a signed statement.

Student Signature: (sign in the same order as above)		Date:
1.	Samuel MALONEY	12/ 6/ 2022
2.	Lining HOU	12/ 6/ 2022
3.	Kane O'Brien	12/ 6/ 2022

University of South Australia
Division of STEM
Machine Learning & Vision Systems (**MFET 4015**)
Group Project

Group Project 2022: MATLAB Image Processing

Students:
Lining HOU
Sam MOLONEY
Kane O'BRIEN

Submission Date: 12/06/22

Supervisors:
Professor Lee Sang-Heon
Md Ahsan Kabir
Rana Islam

Disclaimer

We declare the following to be our own work, unless otherwise referenced, as defined by the University's policy on plagiarism.

Lining Hou (xxxx)

Sam Moloney (xxxxx)

Kane O'Brien (110268293)

Abstract

Here goes an Abstract

Table Of Contents

Disclaimer	i
Abstract	ii
1 Introduction	1
2 Code Listings	3
3 Conclusion	7
4 Appendices	9

List of Figures

List of Tables

1 Introduction

An introduction goes here

Solomon & Breckon (2011) is the reference book we will use.

When performing some communication systems over the radios, it is possible to use multi-rate techniques (Harris 2004, pg. 54). Use of the \citep [pg. x] {label} allows a ‘paragraph citation’

Harris (2021) describes the updated process of multi-rate filtering for highspeed RF. Use of the \citet [pg. x] {label} allows the ‘in text citation’.

You do not have to add anything to the references at the end of the document; it will be automatically updated. You do have to add new sources to /sections/bib_main.bib file. For most things the book mode is satisfactory — use the things in there as examples for URL’s etc. etc.

(Solomon & Breckon 2011)

2 Code Listings

Question 1:

```
%% Question 1:

%initialise environment
clear all
close all
clc

% Part A)
% change this line to swap the image set (1 or 2)
ver = 1;

if ver == 2
    im1 = "../image1a.jpg";
    im2 = "../image2a.jpg";
    ver = "Second";
else
    im1 = "../image1b.jpg";
    im2 = "../image2b.jpg";
    ver = "First";
end

fprintf("Working with the " +ver+ " version.\n\n")

image1 = imread(im1);
image2 = imread(im2);

fprintf("The 'whos' command output:\n")
whos

figure
subplot(2,1,1)
    imshow(im1)
    title("Image 1 – Original")
    impixelinfo
subplot(2,1,2)
    imshow(im2)
    title("Image 2 – Original")
    impixelinfo
pause(1/10)

% Part B)
% Capture values @ 250x,350y
rgb(1:3) = image1(250,350,:);
```

```

gs = image2(250,350);

fprintf("The values of Image 1 250th Row, 350th Column:\nR: %d\nG: %d\nB: %d
        \n\n", rgb(1:3) )
fprintf("The value of Image 2 250th Row, 350th Column:\nGray: %d\n\n", gs)

% Part C
box_dim = 25; % box dimension = 25x25px

% Cant do 25/2 as it is non-integer; we take a floor and ceiling to ensure
% it includes 25; floor will truncate the half, ceiling will round the half
imx1 = 250 - floor(box_dim/2);
imx2 = 250 + ceil(box_dim/2);
imy1 = 350 - floor(box_dim/2);
imy2 = 350 + ceil(box_dim/2);

% Give a new var to preserve original.
img = image1;

% change all three layers to the gs value
img(imx1:imx2,imy1:imy2,:) = gs;
% add the red dot at the central point
img(250,350,:)= cat(3,255,0,0);

figure(2)
imshow(img)
title("Modified image, 25x25px box and red dot centered at 250r 350c")
impixelinfo
pause(1/10)

% Part D
[h,w,d] = size(image1);
% Crop image
img = image1( ceil(h/2)+1:end ,:,:);

figure(3)
imshow(img)
title("Cropped bottom half of the image, same width")
impixelinfo
pause(1/10)

% Part E
img_r = img(:,:,:1);
img_g = img(:,:,:2);
img_b = img(:,:,:3);

figure(4)
subplot(2,2,1)
imshow(img)

```

```

        title("Cropped initial image")
        impixelinfo
    subplot(2,2,2)
        imshow(img_r)
        title("Cropped image; Red layer")
        impixelinfo
    subplot(2,2,3)
        imshow(img_g)
        title("Cropped image; Green layer")
        impixelinfo
    subplot(2,2,4)
        imshow(img_b)
        title("Cropped image; Blue layer")
        impixelinfo
    pause(1/10)

% relaxed, manually observed threshold values
r_th = img_r > 200;
g_th = img_g > 200;
b_th = img_b > 200;

% Part F)
img_th = r_th & b_th;

SE1 = strel('square',15);

img_fill = imfill(img_th, 'holes');
img_erode =imerode(img_fill,SE1);
img_recon = imreconstruct(img_erode,img_fill);

% Double check there was only one; else throw exception error
assert(max(max(bwlabel(img_recon))) == 1, "There was more than one label")

figure(5)
    imshow(img_recon)
    title("Reconstructed, de-noised logo; there is only one object.")
    impixelinfo
pause(1/10)

% Part G)
img_struct = regionprops(img_recon,"BoundingBox");
pos = floor(img_struct.BoundingBox(1:2));
siz = round(img_struct.BoundingBox(3:4));

% Part H)
% Shrink the image to match the bounding box
img2_rz = imresize(image2, [siz(2) siz(1)]); % ensure flip (FLIPLR) the
matrix to account for the xy => yx imtool thing

```

```
% Part I)
% Replace the contents of the bounding box
img_r(pos(2):pos(2)+siz(2)-1, pos(1):pos(1)+siz(1)-1) = img2_rz;
img_g(pos(2):pos(2)+siz(2)-1, pos(1):pos(1)+siz(1)-1) = img2_rz;
img_b(pos(2):pos(2)+siz(2)-1, pos(1):pos(1)+siz(1)-1) = img2_rz;
img = cat(3,img_r, img_g, img_b);

figure(6)
imshow(img)
title("Replaced Logo, Cropped image")
impixelinfo
pause(1/10)

% Part J)
% Replace the top half image

image_top = image1( 1:floor(h/2) ,:,:);
image_final = [image_top; img];

figure(7)
imshow(image_final)
title("Final Image with Logo Replaced")

assert(sum((size(image1)) == (size(image_final))))== 3, "Major mistakes were
made; the size is probably not the same")
fprintf("The sizes of each image were checked and they are the same size of
:\n%d x %d x %dpixels\n\nQ1. Complete \n\n", size(image_final))
```

3 Conclusion

Write a conclusion here :)

References

- Harris, FJ 2004, *Multirate Signal Processing for Communication Systems*, 1st edn, Prentice Hall PTR, Upper Saddle River, New Jersey.
- Harris, FJ 2021, *Multirate Signal Processing for Communication Systems*, 2nd edn, River Publishers, Gistrup, Denmark.
- Solomon, C & Breckon, T 2011, *Fundamentals of Digital Image Processing: a Practical Approach with Examples in Matlab*, 1st edn, John Wiley and Sons, West Sussex, United Kingdom.

4 Appendices