Lab1:

A is an array. A(find(A < 0)) = 0

&, |, ~ for **and**, **or**, and **not**

if x > 10

x = x + 1; % **no ++ or even +=** in MATLAB

else

x = x – 1;

end

j=1;

for i = 1:10

j = j\*i;

end

>> img = imread(‘image\foo.jpg’);

>>img %to see the pixel values.

>> whos %to see its size. What are the 3 dimensions of your image?

>>imshow(img) %to see the image.

>>imtool(img)

>> red = img(:,:,1);

>> imtool(red);

avg = uint8((double(red) + double(green) + double(blue))/3);

Lab2:

imresize(img, 0.5) % rescales (shrinks) to ½ size

imresize(img, 0.5, ‘bicubic’); % same, but higher quality

Look at the distribution of pixel values of each band individually using **imhist()**.

>> imhist(v); % to look at the distribution of intensities, for example.

Add a mask to a picture

%% Real image

img = imread('../../Images/covert.jpg');

imtool(img);

mask = ones(size(img, 1), size(img, 2));

% bgIdx = find(img(:,:,3) > 140);

bgIdx = find(img(:,:,1) < 130 & ...

img(:,:,2) > 110 & img(:,:,2) < 160 & ...

img(:,:,3) > 140);

mask(bgIdx) = 0;

imtool(mask);

imwrite(mask, 'raw\_mask.png');

face = repmat(mask, [1,1,3]) .\* double(img);

imtool(uint8(face))

morphology:

se = strel('square',3);

erodedImg = imerode(img,se);

se = strel('square',5);

dilatedImg = imdilate(img,se);

closedImg = imclose(img, strel('square', 11));

Lab3:

filter2(matrix, img)

A.\*B

function retValue = foo(param1, param2, …)

… %calculations go here…

retValue = …;

function [retValue1, retValue2, …] = foo(param1, param2, …)

atan2