Joanne Kwon

PID: A15359545

**Part I**

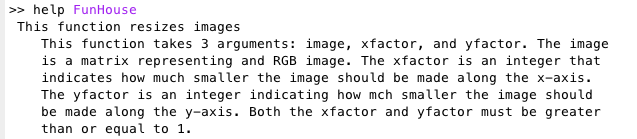
QUESTION #1

* I see a small box on the screen. This is because imshow brings up a 4x5x3 uint8 array of the image.

QUESTION #2

* When I typed in ‘help FunHouse’ it outputted the summary and explanation that I had commented in the function.



QUESTION #3

Code:

% Name: Joanne Kwon

% The function FunHouse resizes images.

% This function takes 3 arguments: image, xfactor, and yfactor. The image

% is a matrix representing and RGB image. The xfactor is an integer that

% indicates how much smaller the image should be made along the x-axis.

% The yfactor is an integer indicating how much smaller the image should

% be made along the y-axis. Both the xfactor and yfactor must be greater

% than or equal to 1.

function [imagematrix] = FunHouse(image, xfactor, yfactor)

imagematrix = (image(floor(1/yfactor):1,(1:floor(1/xfactor))));

end

QUESTION #4

* When executing “FunHouse();” with no inputs there was an error because there were not enough input arguments.

QUESTION #5

* When using imagematrix instead of image and leaving the argument list on the first line the same there is an error output because imagematrix is an undefined variable.

QUESTION #6

* I conclude that changing the return variable name from finalmatrix to imagematrix will simply leave the variable finalmatrix and only alter the values in the new variable, imagematrix, because imagematrix is the new return variable.



QUESTION #7

* It might not always be a good thing for scripts to have complete access to the workspace because it can modify/alter the values in the workspace even when you don’t want it to.

**Part II**

**ExtractLayer**

Code:

% Name: Joanne Kwon

% The function "ExtractLayer" returns a matrix representing an image with only the requested color layer.

% The image is a matrix that represents an RGB image in the usual RGB image format. The Layer % is a number taking values 1, 2, or 3 representing the red, green, and blue layers respectively



function [finalmatrix] = ExtractLayer(image, layer)



red = image(:,:,1);

green = image(:,:,2);

blue = image(:,:,3);

a = zeros(size(image, 1), size(image, 2));



if (layer == 1)

finalmatrix = cat(3, red, a, a);

elseif (layer == 2)

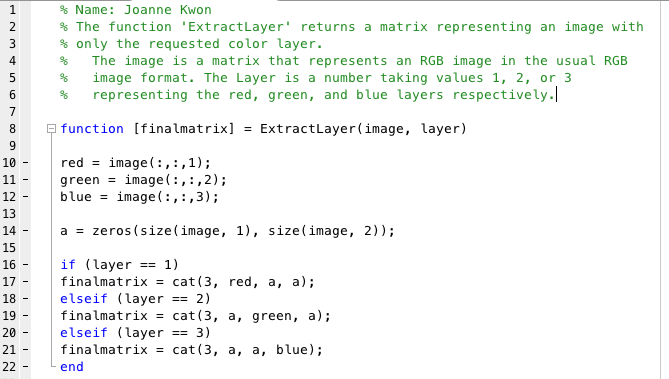
finalmatrix = cat(3, a, green, a);

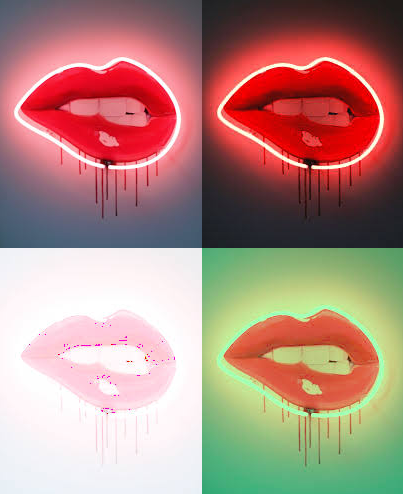
elseif (layer == 3)

finalmatrix = cat(3, a, a, blue);

end







**TileImage**

Code:

% Name: Joanne Kwon

% The function 'TileImage' takes four arguments (image1, image2, image3,

% image4) and returns a matrix representing a tiled version of the input

% images.

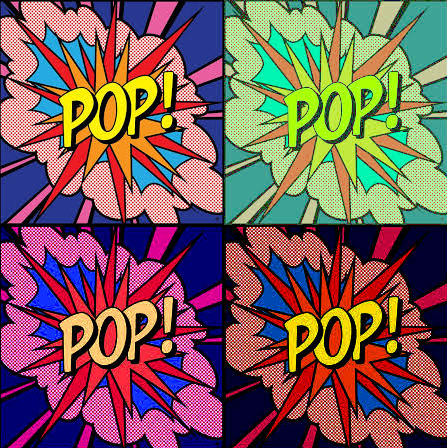
% The arguments are matrices that represents an RGB image in the usual

% RGB image format. It can be assumed that all four image matrices will

% have the exact same dimensions.

function [NewImage] = TileImage(image1, image2, image3, image4)

rows = size (image1, 1);

columns = size (image1, 2);

layers = size (image1, 3);

NewImage = uint8(zeros(2.\*rows,2.\*columns,layers));

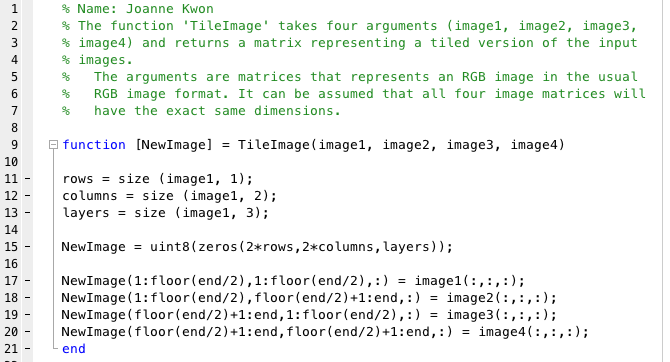
NewImage(1:floor(end/2),1:floor(end/2),:) = image1(:,:,:);

NewImage(1:floor(end/2),floor(end/2)+1:end,:) = image2(:,:,:);

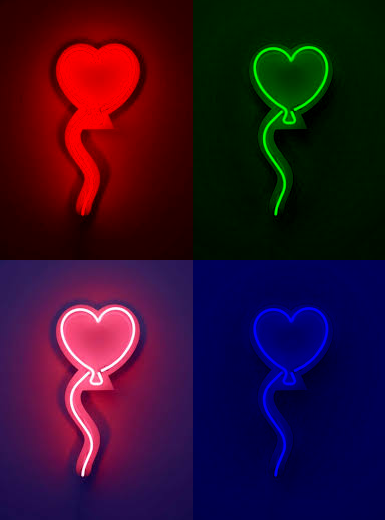
NewImage(floor(end/2)+1:end,1:floor(end/2),:) = image3(:,:,:);

NewImage(floor(end/2)+1:end,floor(end/2)+1:end,:) = image4(:,:,:);

end



**MakeWarhol**

Code:

% Name: Joanne Kwon

% The function 'MakeWarhol' uses the 'ExtractLayer' and 'TileImage'

% functions to create and display a tiled image with four variation of the

% image found in filename.

% The function uses 'ExtractLayer' to produce some different versions of

% the image and stixhes them together with 'TileImage' and displays the

% results. The top left image is the red version, the top right image is

% the green version, the bottom left image is the original image and the

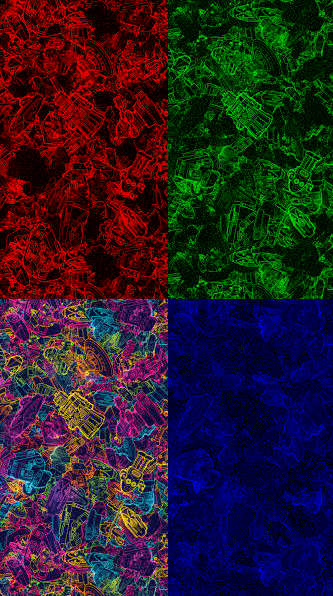
% bottom right image is the blue version.

function [NewDisplay] = MakeWarhol(filename)

red = filename (:, :, 1);

green = filename (:, :, 2);

blue = filename (:, :, 3);

NewDisplay = uint8(zeros(2\*size(filename, 1), 2\*size(filename,2),3));

NewDisplay(1:floor(end/2),1:floor(end/2),1) = red;

NewDisplay(1:floor(end/2),floor(end/2)+1:end,2) = green;

NewDisplay(floor(end/2)+1:end,1:floor(end/2),:) = filename;

NewDisplay(floor(end/2)+1:end,floor(end/2)+1:end,3) = blue;

end

