Assessment 2C: Image comparison code workbook

Nikki Fitzherbert jc697513

Comparison between all images

Perfect similarity scores are expected for the pairs comparing themselves. The two apple images received a relatively high similarity score of 0.7125 according to the *compareImages* function and the two hammer images should receive a similar value.

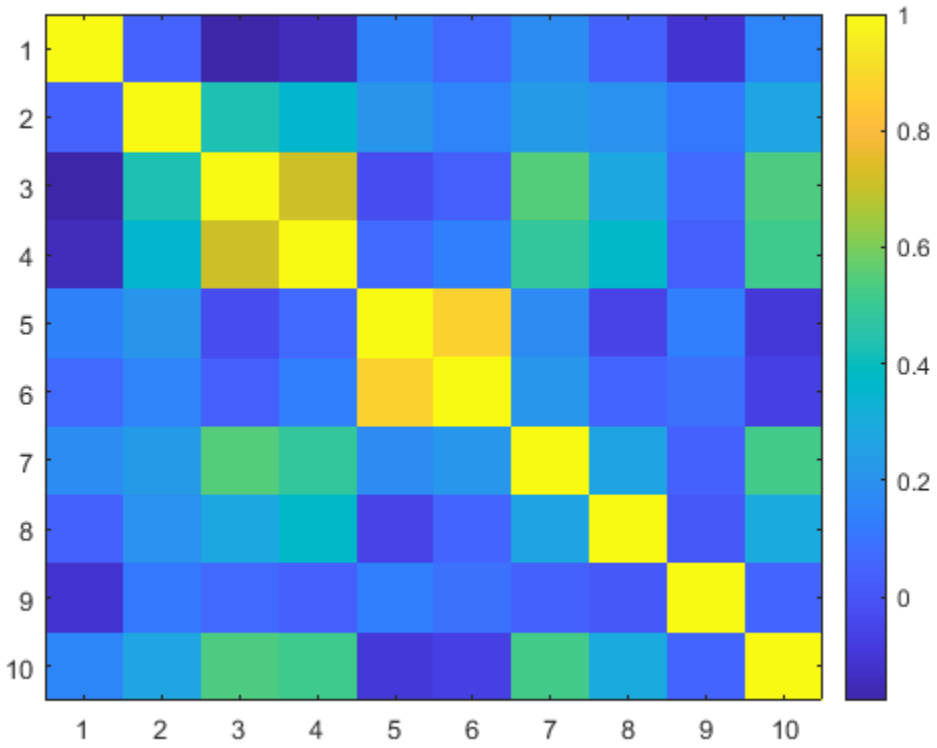
Below is the complete predicted order of image pairs from highest to lowest similarity. Note that there are only 55 pairs as it did not matter which order the pair was in. For example, the similarity score for Hat-Teddy was the same as the similarity score for Teddy-Hat.:

Glass-Glass
Heart-Heart
Apple_13-Apple_13
Apple_9-Apple_9
Hammer_12-Hammer_12
Hammer_4-Hammer_4
Hat-Hat
Imfish-Imfish
Spring-Spring
Teddy-Teddy
Hammer_12-Hammer_4
Apple_13-Apple_9
Glass-Hat
Heart-Apple_13
Heart-Apple_9
Imfish-Teddy
Heart-Hammer_4
Heart-Hammer_12
Apple_9-Hat
Apple_13-Hat
Hammer_12-Imfish
Hammer_12-Teddy
Glass-Hammer_4
Hammer_4-Imfish
Hammer_4-Teddy
Glass-Hammer_12
Hat-Teddy

Apple_9-Teddy
Apple_13-Teddy
Glass-Teddy
Spring-Teddy
Imfish-Spring
Glass-Heart
Glass-Apple_9
Apple_9-Hammer_4
Hammer_4-Hat
Glass-Apple_13
Apple_13-Hammer_4
Apple_9-Hammer_12
Hammer_12-Hat
Heart-Hat
Heart-Teddy
Apple_13-Hammer_12
Hat-Imfish
Hat-Spring
Glass-Spring
Heart-Spring
Hammer_4-Spring
Hammer_12-Spring
Apple_9-Spring
Glass-Imfish
Heart-Imfish
Apple_13-Imfish
Apple_13-Spring
Apple_9-Imfish

```
%% Mass comparison
% We can now compare all of the test images to one another.

% We first obtain the full file list
filelist = dir('*.png')
N = length(filelist)

% filenames = {}
 for i = 1:length(filelist)
     filenames{i} = filelist(i).name;
 end

% We will store each comparison into a matrix, so that each row/column
% corresponds to a particular image.
mat = zeros(N);

% We now loop over all of the files twice, so that we go through each
% element of the matrix.
for iA = 1:N
    fileA = filelist(iA);
    imA = imread(fileA.name);

    for iB = iA:N
        fileB = filelist(iB);

        imB = imread(fileB.name);

        val = compareImages(imA,imB);

        mat(iA,iB) = val;
        mat(iB,iA) = mat(iA,iB);
    end
end

%% Mass comparison results
% After obtaining the comparison between all of the images, we can plot
% this.

imagesc(mat)
colorbar

% We also output all of the filenames and their corresponding number.
for i = 1:N
    disp(['i = ', num2str(i), ' is ', filelist(i).name])
end
```

```
i = 1 is Glas-1.png
i = 2 is Heart-2.png
i = 3 is apple-13.png
i = 4 is apple-9.png
i = 5 is hammer-12.png
i = 6 is hammer-4.png
i = 7 is hat-19.png
i = 8 is lmfish-11.png
i = 9 is spring-9.png
i = 10 is teddy-04.png
```



The array is symmetric across the diagonal, which shows the (perfect) similarity scores of the images compared to themselves. Therefore, the discussion of the patterns in the scores will concentrate on the lower half. There are also two other lighter squares that indicate the high similarities of the two apple images and two hammer images as was expected. The majority of the rest of the array received relatively low similarity scores except for three images paired with the teddy bear image (image/row 10), two with the hat (image/row 7) and the heart-apple-13 (image/row 13).

The patterns displayed are mostly as expected, except the high similarity scores for the teddy-apple pairs , and the apple_9-imfish pair was a surprise. On the opposite on of the spectrum, the low similarity scores for such pairings as the hammers-teddy and glass-hat were also unexpected.