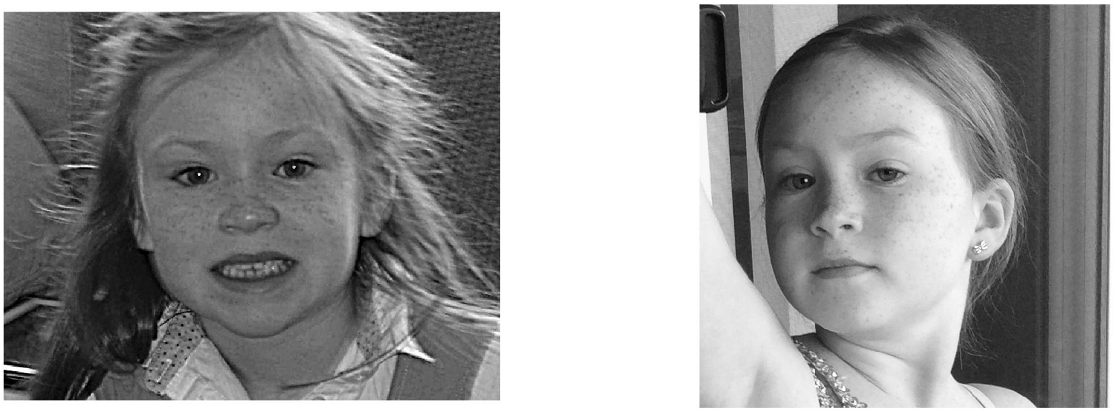
Live Script for Testing Facial Expression Recognition Algorithm

Process for Facial Expression Recognition is as follows:

- Input Labeled Images into workspace and categorize them based on 5 emotions: Happy, Sad, Angry, Surprised, Fear, and Neutral
- Make images Monochrome if they are RGB images, and intensity scale the images to have pixel values between 0 and 1
- Preprocess the images by resizing them and then cropping them to be a determined M x N sized image
- Extract the two main areas of interest of the faces, which are the Eye/Eyebrow and Mouth regions
- Once regions are extracted from original image, use the 2D Discrete Cosine Transform to extract low frequency coefficients that happen to capture the most relevant info about the facial expressions
- 15 LF coefficients from both the Eye/Mouth region of one image will be combined into a single array and used to be input into the Classification Learner App inside MATLAB
- Once the Neural Network is trained using enough images, output the trained model and instantiate it as a module within the custom GUI for our project
- Ideally, the GUI will allow the user to upload a new raw image and will follow the exact steps laid out above and result in a classification output image of one of the 5/6 emotions will some degree of confidence.
- The CI we are hoping to achieve should be around 95%, but if we run out of time, we will present methods of optimizing the model to a obtain a higher degree of CI

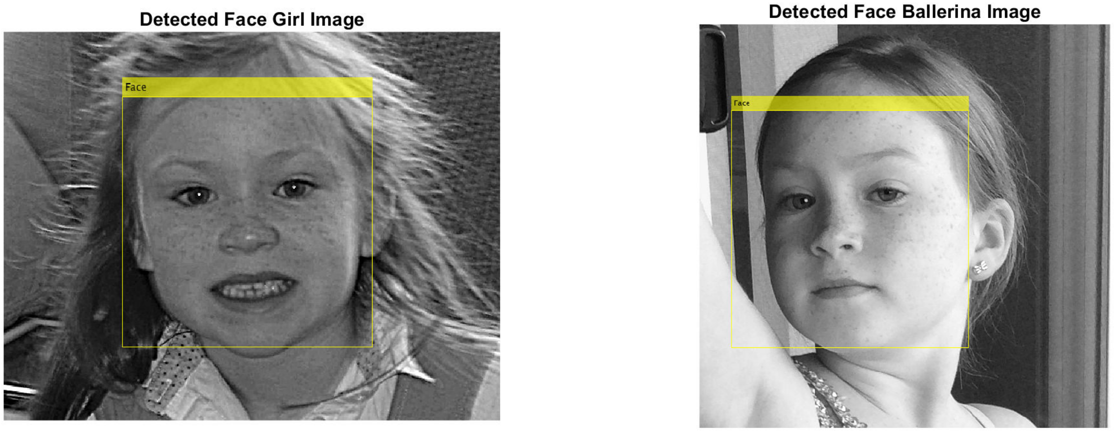
Test Preprocessing Images by Cropping Faces

```
clc; close all; clear;
f1 = intensityScaling(imread("girl.tif"));
f2 = intensityScaling(rgb2gray4e(imread('ballerina.tif')));
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(f1)
subplot(1,2,2), imshow(f2)
```



```
faceDetector = vision.CascadeObjectDetector; % Default: finds faces
bboxes1 = step(faceDetector, f1); % Detect faces
```

```
bboxes2 = step(faceDetector, f2); % Detect faces
% Annotate detected faces
IFaces1 = insertObjectAnnotation(f1, 'rectangle', bboxes1, 'Face');
IFaces2 = insertObjectAnnotation(f2, 'rectangle', bboxes2, 'Face');
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(IFaces1), title('Detected Face Girl Image');
subplot(1,2,2), imshow(IFaces2), title('Detected Face Ballerina Image');
```

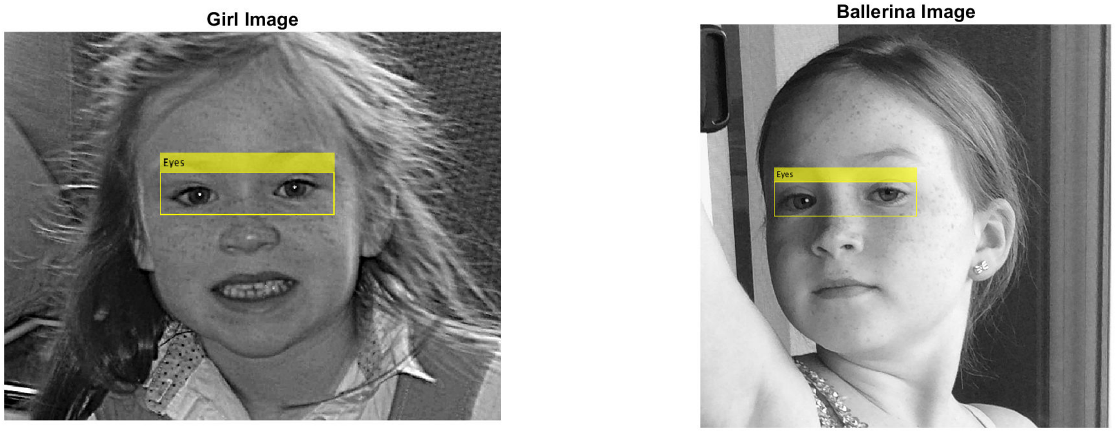


```
croppedf1 = imcrop(f1,bboxes1);
croppedf2 = imcrop(f2,bboxes2);
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(croppedf1), title('Cropped Girl Face');
subplot(1,2,2), imshow(croppedf2), title('Cropped Ballerina Face');
```

Test Preprocessing Images by Cropping Eye and Mouth Regions

```
clc; close all; clear;
girl = imread("girl.tif");
ballerina = rgb2gray4e(imread('ballerina.tif'));
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(girl), title('Girl Image');
subplot(1,2,2), imshow(ballerina), title('Ballerina Image');
```

```
%Detect Eyes
EyeDetect = vision.CascadeObjectDetector('EyePairBig','MergeThreshold',7);
eyes_bboxes1 = step(EyeDetect,girl);
eyes_bboxes2 = step(EyeDetect,ballerina);
% Annotate detected Eye Region
IFaces1 = insertObjectAnnotation(girl, 'rectangle', eyes_bboxes1, 'Eyes');
IFaces2 = insertObjectAnnotation(ballerina, 'rectangle', eyes_bboxes2, 'Eyes');
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(IFaces1), title('Girl Image');
subplot(1,2,2), imshow(IFaces2), title('Ballerina Image');
```



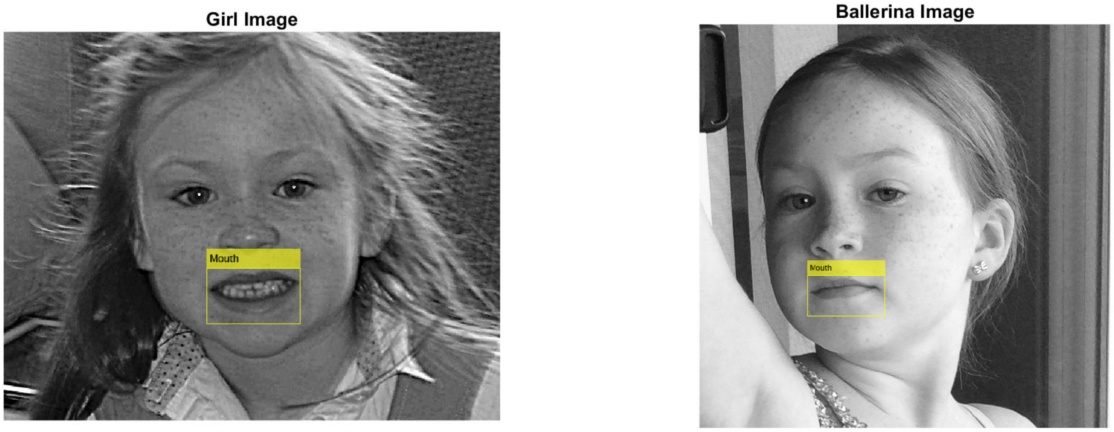
```
% Crop Eye Regions
eyes_girl = imcrop(girl,eyes_bboxes1);
eyes_ballerina = imcrop(ballerina,eyes_bboxes2);

figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(eyes_girl), title('Eye Region of Girl Image');
subplot(1,2,2), imshow(eyes_ballerina), title('Eye Region of Ballerina Image');
```

Eye Region of Girl Image

Eye Region of Ballerina Image

```
% Detect Mouth
MouthDetect = vision.CascadeObjectDetector('Mouth','MergeThreshold',150);
mouth_bboxes1 = step(MouthDetect,girl);
mouth_bboxes2 = step(MouthDetect,ballerina);
mouth_bboxes1(1) = mouth_bboxes1(1) + 5;
mouth_bboxes2(1) = mouth_bboxes2(1) - 15;
mouth_bboxes2(3) = mouth_bboxes2(3) + 15;
% Annotate detected Mouth Region
IFaces1 = insertObjectAnnotation(girl, 'rectangle', mouth_bboxes1, 'Mouth');
IFaces2 = insertObjectAnnotation(ballerina, 'rectangle', mouth_bboxes2, 'Mouth');
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(IFaces1), title('Girl Image');
subplot(1,2,2), imshow(IFaces2), title('Ballerina Image');
```

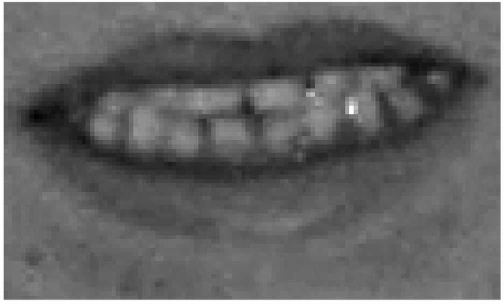


Girl Image

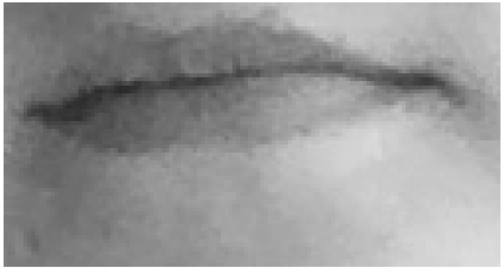
Ballerina Image

```
% Crop Mouth Region
mouth_girl = imcrop(girl,mouth_bboxes1);
mouth_ballerina = imcrop(ballerina,mouth_bboxes2);
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,2,1), imshow(mouth_girl), title('Detected Mouth Region of Girl Image');
subplot(1,2,2), imshow(mouth_ballerina), title('Detected Mouth Region of Ballerina Image');
```

Detected Mouth Region of Girl Image



Detected Mouth Region of Ballerina Image



Perform 2D Discrete Cosine Transform to extract DCT coefficients

```
DCT_mouth_girl = dct2(mouth_girl);
DCT_eyes_girl = dct2(eyes_girl);
DCT_mouth_ballerina = dct2(mouth_ballerina);
DCT_eyes_ballerina = dct2(eyes_ballerina);
figure('Units','inches','Position',[0,0,12,4]);
subplot(2,2,1); imshow(DCT_mouth_girl)
subplot(2,2,2); imshow(DCT_eyes_girl)
subplot(2,2,3); imshow(DCT_mouth_ballerina)
subplot(2,2,4); imshow(DCT_eyes_ballerina)
```

Try Extracting a 4 x 4 grid of the lowest DCT coefficients, so 16 low frequency values, from both Eye and Mouth Regions for both images

```
Mouthtable = zeros(2,16);
Mouthtable(1,1:4) = DCT_mouth_girl(1,1:4);
Mouthtable(1,5:8) = DCT_mouth_girl(2,1:4);
Mouthtable(1,9:12) = DCT_mouth_girl(3,1:4);
Mouthtable(1,13:16) = DCT_mouth_girl(4,1:4);
Mouthtable(2,1:4) = DCT_mouth_ballerina(1,1:4);
Mouthtable(2,5:8) = DCT_mouth_ballerina(2,1:4);
Mouthtable(2,9:12) = DCT_mouth_ballerina(3,1:4);
Mouthtable(2,13:16) = DCT_mouth_ballerina(4,1:4);
Mouthtable
```

```
Mouthtable = 2×16
   32.3319   -0.1464    0.6983    0.0281   -1.5206    1.0276   -0.3057   -0.3571 ···
   53.5743  -10.8330   -0.4402    1.6441   -4.1551    1.1754    0.4023    1.0246
```

```
Eyetable = zeros(2,16);
Eyetable(1,1:4) = DCT_eyes_girl(1,1:4);
Eyetable(1,5:8) = DCT_eyes_girl(2,1:4);
Eyetable(1,9:12) = DCT_eyes_girl(3,1:4);
Eyetable(1,13:16) = DCT_eyes_girl(4,1:4);
Eyetable(2,1:4) = DCT_eyes_ballerina(1,1:4);
Eyetable(2,5:8) = DCT_eyes_ballerina(2,1:4);
Eyetable(2,9:12) = DCT_eyes_ballerina(3,1:4);
Eyetable(2,13:16) = DCT_eyes_ballerina(4,1:4);
Eyetable
```

```
Eyetable = 2×16
   40.7387   -1.5540   -1.2392   -1.8001   -1.8380    2.6869   -1.2690   -0.3855 ···
   54.2873  -13.7725   -6.5982   -3.4001   -0.0134    3.9317    0.3003   -2.6606
```

```
DCT_table = horzcat(Eyetable,Mouthtable);
DCT_table = array2table(DCT_table);
DCT_table.Properties.VariableNames = {'Eye C1' 'Eye C2' 'Eye C3' 'Eye C4' 'Eye C5' 'Eye C6' 'Ey
DCT_table.EmotionClass = cell(2,1);
DCT_table.EmotionClass(1) = {'Girl'};
DCT_table.EmotionClass(2) = {'Ballerina'};
DCT_table.EmotionClass = categorical(DCT_table.EmotionClass);
DCT_table
```

DCT_table = 2×33 table

...

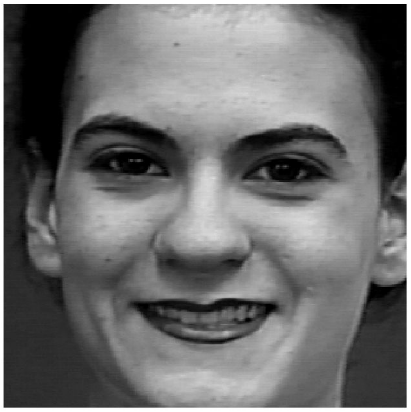
	Eye C1	Eye C2	Eye C3	Eye C4	Eye C5	Eye C6	Eye C7	Eye C8
1	40.7387	-1.5540	-1.2392	-1.8001	-1.8380	2.6869	-1.2690	-0.3855
2	54.2873	-13.7725	-6.5982	-3.4001	-0.0134	3.9317	0.3003	-2.6606

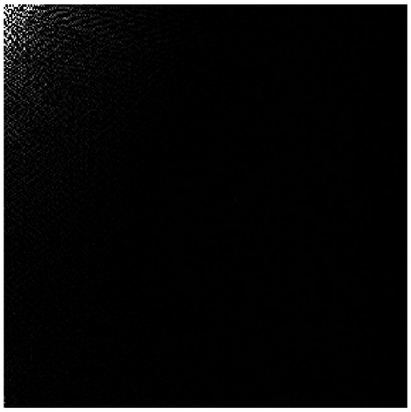
Extract DCT Coefficients from 5 basic emotion images and input into Classification Learner

Start first by extracting the face DCT coefficients

```
n = 5;
a = imread('happy.png');
b = rgb2gray4e(imread('sadness.png'));
c = imread('anger.png');
d = imread('disgust.png');
e = rgb2gray4e(imread('fear.png'));
```

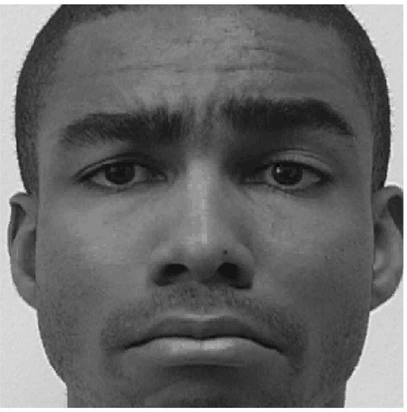
```
featureFaceTable = zeros(5,n^2);
featureFaceTable(1,1:n^2) = extractDCT_Face(a,n);
```

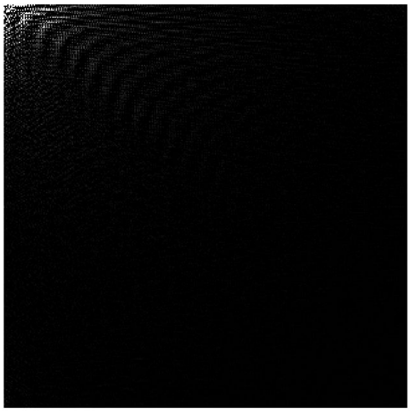




```
faceTable = 5×5
  114.5032  -16.8366  -39.1046    4.0799   -8.8672
    2.6326   -1.6787  -14.8329    0.7027   -1.1564
   -2.6500    1.2791  -18.6784    6.3757    0.9617
    6.1945    0.2373    1.8834   -0.0128   -5.9618
    7.6905    2.7650   -4.5901   -1.8667    7.3904
featureVector = 1×25
  114.5032  -16.8366    2.6326   -2.6500   -1.6787  -39.1046    4.0799  -14.8329···
```

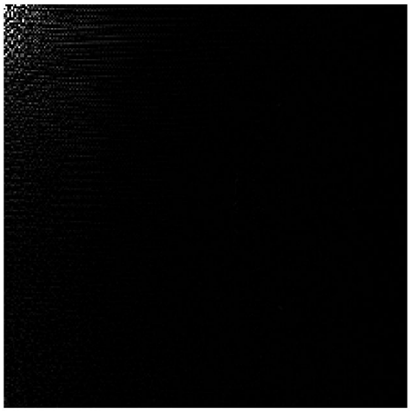
```
featureFaceTable(2,1:n^2) = extractDCT_Face(b,n);
```





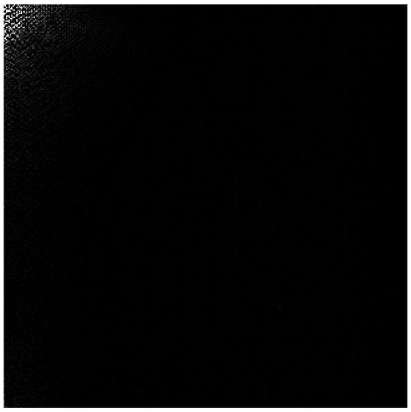
```
faceTable = 5×5
  129.5790   10.1565   22.9195   -2.2339   22.4817
   -1.6178   -1.3317  -12.6173    0.2234    2.1732
    9.7803   -1.7643   14.2019   -2.7569   11.2755
   10.2399    0.3246   -8.1665    0.6256   -4.7435
    9.3898   -0.2997    0.5805    1.3034    1.1468
featureVector = 1×25
  129.5790   10.1565   -1.6178    9.7803   -1.3317   22.9195   -2.2339  -12.6173···
```

```
featureFaceTable(3,1:n^2) = extractDCT_Face(c,n);
```



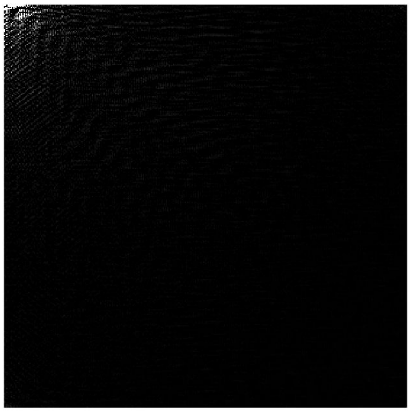
```
faceTable = 5×5
   42.0373   -1.7756  -17.2929   -0.2979   -3.0955
    1.7559    1.1529   -7.4732   -0.2451    3.6238
   -9.3822    0.8487   -3.8682   -1.1028    2.6372
    4.5326    0.9614   -2.7685    0.5907   -1.1139
    3.1814   -0.6180    0.5180   -0.0229    1.2644
featureVector = 1×25
   42.0373   -1.7756    1.7559   -9.3822    1.1529  -17.2929   -0.2979   -7.4732 ···
```

```
featureFaceTable(4,1:n^2) = extractDCT_Face(d,n);
```



```
faceTable = 5×5
   77.7551    5.4258  -18.5890   -4.1508    0.6488
  -10.4590    0.9683  -22.7840   -0.4336    0.8008
  -10.2922   -2.5886    3.3060   -2.4967    8.8883
    1.0418    1.3296   -4.0609   -0.0894   -3.9038
    4.3298   -2.2042    4.5851    0.7449    7.1422
featureVector = 1×25
   77.7551    5.4258  -10.4590  -10.2922    0.9683  -18.5890   -4.1508  -22.7840 ···
```

```
featureFaceTable(5,1:n^2) = extractDCT_Face(e,n);
```



```
faceTable = 5×5
  170.7177   18.9118   -2.5091    4.1676    5.3726
   -8.5172   -3.3257  -19.3008    4.6265    4.7070
   -4.0495    1.5243    6.5623    0.0277   19.4458
    8.0757    2.0365    8.4167    2.7185    6.0359
    3.6243    1.4406    1.5453    0.3816   11.1021
featureVector = 1×25
  170.7177   18.9118   -8.5172   -4.0495   -3.3257   -2.5091    4.1676  -19.3008 ···
```

```
featureFaceTable = array2table(featureFaceTable);
for i = 1:n^2
    fname = ['B',num2str(i)];
    featureFaceTable.Properties.VariableNames(i) = {fname};
end

featureFaceTable.EmotionClass = cell(5,1);
featureFaceTable.EmotionClass = {'happy'; 'sad'; 'anger'; 'disgust'; 'fear'}
```

featureFaceTable = 5×26 table

...

	B1	B2	B3	B4	B5	B6	B7	B8	B9
1	114.5032	-16.8366	2.6326	-2.6500	-1.6787	-39.1046	4.0799	-14.8329	1.2791
2	129.5790	10.1565	-1.6178	9.7803	-1.3317	22.9195	-2.2339	-12.6173	-1.7643
3	42.0373	-1.7756	1.7559	-9.3822	1.1529	-17.2929	-0.2979	-7.4732	0.8487
4	77.7551	5.4258	-10.4590	-10.2922	0.9683	-18.5890	-4.1508	-22.7840	-2.5886
5	170.7177	18.9118	-8.5172	-4.0495	-3.3257	-2.5091	4.1676	-19.3008	1.5243

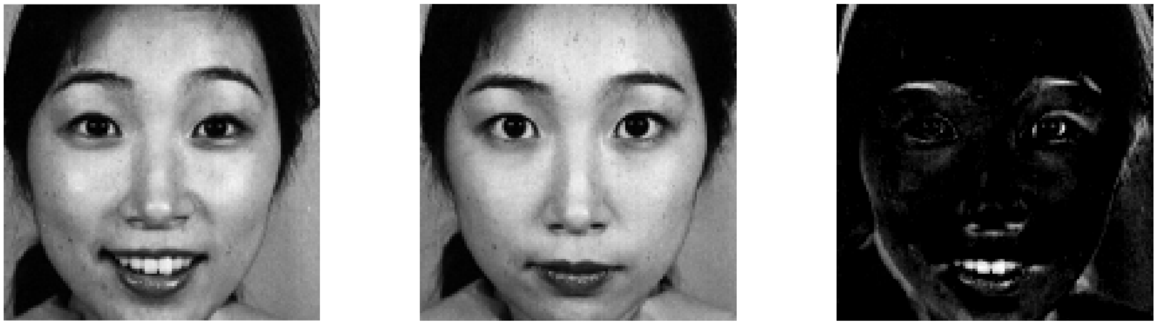
```
featureFaceTable.EmotionClass = categorical(featureFaceTable.EmotionClass);
```

```
a = [1,2,4; 3,5,7; 6,8,9;];
b = zigzag(a)
```

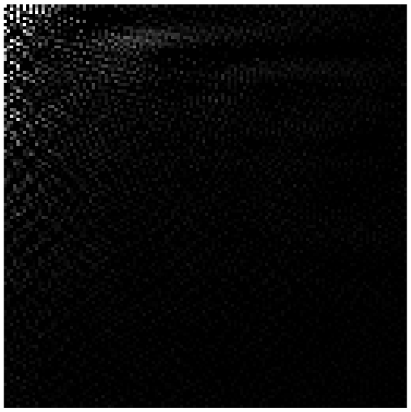
```
b = 1×9
     1     2     3     6     5     4     7     8     9
```

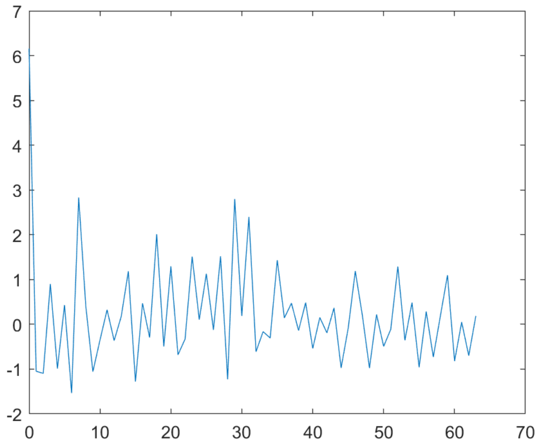
Create a Difference Image for DCT coefficient extraction

```
n = 8;
f = imread('HA2.tiff');
g = imread('NE2.tiff');
% Use Computer Vision Toolbox Functions to extract Face crop image
FaceDetect = vision.CascadeObjectDetector;
face_bboxes1 = step(FaceDetect, f);
face_bboxes2 = step(FaceDetect, g);
% Crop face from original image
face1 = imcrop(f,face_bboxes1);
face2 = imcrop(g,face_bboxes2);
face1 = imresize(face1,[128,128]);
face2 = imresize(face2,[128,128]);
difface = intensityScaling(imsubtract(face1,face2));
figure('Units','inches','Position',[0,0,12,4]);
subplot(1,3,1),imshow(face1);
subplot(1,3,2),imshow(face2);
subplot(1,3,3),imshow(difface);
```



```
DCT_face = dct2(difface);
% Extract an n x n block from the low frequencies of the DCT coefficients
facetable = DCT_face(1:n,1:n);
% Once the DCT Matrix is extracted, calculate the mean and standard
% deviation for the row-column-diagonal of the matrix and result in a
% feature vector for input into a KNN model
featureVector = zigzag(facetable);
% Generate plot of Image with DCT-II transform image
figure('Units','inches','Position',[0,0,12,4])
subplot(1,2,1),imshow(DCT_face);
subplot(1,2,2), plot(0:length(featureVector)-1,featureVector);
```




7
6
5
4
3
2
1
0
-1
-2
0
10
20
30
40
50
60
70