A

First we must load the images:

```
img_a = imread("./imgs/a_visbileblue.jpg");
img_b = imread("./imgs/b_visiblegreen.jpg");
img_c = imread("./imgs/c_visiblered.jpg");
img_d = imread("./imgs/d_nearinfrared.jpg");
img_e = imread("./imgs/e_middleinfrared.jpg");
img_f = imread("./imgs/f_thermalinfrared.jpg");
```

We are assuming a grayscale image, so convert them:

```
img_a = rgb2gray(img_a);
img_b = rgb2gray(img_b);
img_c = rgb2gray(img_c);
img_d = rgb2gray(img_d);
img_e = rgb2gray(img_e);
img_f = rgb2gray(img_f);
```

Then we convert them to vectors:

```
vector_a = reshape(img_a, [256*256,1]);
vector_b = reshape(img_b, [256*256,1]);
vector_c = reshape(img_c, [256*256,1]);
vector_d = reshape(img_d, [256*256,1]);
vector_e = reshape(img_e, [256*256,1]);
vector_f = reshape(img_f, [256*256,1]);
```

Note that these vectors are uint8 due to being images, but we need floating point values - let's convert them to doubles.

```
vector_a = double(vector_a);
vector_b = double(vector_b);
vector_c = double(vector_c);
vector_d = double(vector_d);
vector_e = double(vector_e);
vector_f = double(vector_f);
```

Finally we can combine them to a gigantic vector which will be 6x65,536...

```
vector = [vector_a vector_b vector_c vector_d vector_e vector_f ]'
```

```
vector = 6×65536
   255   249   255   255   254   250   251   255   253   255   253   254   254 ···
   139   126   131   131   127   130   127   127   111   137   122   119   130
    87    80    62    54    89   113    83    53    43    62    66    64    69
   208   175   203   187   203   227   206   194   200   195   200   205   196
   194   160   179   162   171   187   175   169   171   171   168   171   173
   146    67    70    59    75   111    98    61    42    60    63    55    69
```

Let's find the mean matrix:

```
mean_matrix = mean(vector,2)
```

```
mean_matrix = 6×1
  108.8263
  110.2227
  108.4957
  157.1799
  141.9874
  109.6162
```

Now we can find the covariant matrix Cx:

```
Cx = zeros(6);
for i=1:256*256
    tmp = vector(:,i) - mean_matrix;
    Cx = Cx + (1/(256*256))*(tmp*tmp.');
end
Cx
```

```
Cx = 6×6
10^3 ×
    1.4610    1.2925    1.1335   -0.3030    0.4125    1.0947
    1.2925    1.2832    1.1826   -0.2736    0.4329    1.1341
    1.1335    1.1826    1.3830   -0.3537    0.3605    1.3074
   -0.3030   -0.2736   -0.3537    0.8365    0.3419   -0.3855
    0.4125    0.4329    0.3605    0.3419    0.5007    0.3596
    1.0947    1.1341    1.3074   -0.3855    0.3596    1.3695
```

Now we find the eigenvalues and eigenvector of our calculated Cx

```
[eigenvectors, eigenvalues] = eig(Cx)
```

```
eigenvectors = 6×6
    0.3125   -0.4534   -0.0681   -0.6654   -0.0827    0.4925
   -0.6413    0.5152   -0.0615   -0.2757   -0.0969    0.4838
    0.4548    0.2810   -0.5261    0.4365    0.0457    0.4947
   -0.2122   -0.2711   -0.4210    0.1286   -0.8179   -0.1367
    0.3710    0.3267    0.6487    0.0682   -0.5535    0.1542
   -0.3187   -0.5195    0.3415    0.5192    0.0795    0.4859
eigenvalues = 6×6
10^3 ×
    0.0373         0         0         0         0         0
         0    0.0656         0         0         0         0
         0         0    0.0914         0         0         0
         0         0         0    0.4151         0         0
         0         0         0         0    1.0620         0
         0         0         0         0         0    5.1623
```

B

First, we must sort the eigenvectors in descending order of importance/value:

```
[eigenvalues_ordered, indexes] = sort(diag(eigenvalues), 'descend');
eigenvalues = eigenvalues(indexes, indexes);
eigenvectors = eigenvectors(:,indexes);

A = eigenvectors.'
```

```
A = 6×6
    0.4925    0.4838    0.4947   -0.1367    0.1542    0.4859
   -0.0827   -0.0969    0.0457   -0.8179   -0.5535    0.0795
   -0.6654   -0.2757    0.4365    0.1286    0.0682    0.5192
   -0.0681   -0.0615   -0.5261   -0.4210    0.6487    0.3415
   -0.4534    0.5152    0.2810   -0.2711    0.3267   -0.5195
    0.3125   -0.6413    0.4548   -0.2122    0.3710   -0.3187
```

```
y = A*(vector - mean_matrix)
```

```
y = 6×65536
   94.0420   42.2146   39.2458   29.5079   51.3708   79.4052   58.3072   28.1726 ···
  -83.3214  -42.3525  -77.3355  -56.0788  -70.8065  -95.2957  -73.6726  -65.1778
  -85.6066 -128.6614 -135.4352 -147.8549 -119.8312  -84.6536 -107.8548 -144.7726
   24.3637   -5.8912    4.4250    0.5849  -12.9484  -12.9167   -0.4000    3.6340
  -73.1711  -40.2386  -48.3822  -46.1326  -47.6136  -57.4904  -59.3939  -49.1239
   14.3625   37.2066   27.8392   24.7946   37.8113   34.9237   27.6623   27.3792
```

```
Cy = A* Cx * A.'
```

```
Cy = 6×6
10^3 ×
    5.1623    0.0000    0.0000         0   -0.0000   -0.0000
    0.0000    1.0620   -0.0000    0.0000    0.0000    0.0000
   -0.0000   -0.0000    0.4151    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000    0.0914   -0.0000   -0.0000
   -0.0000    0.0000    0.0000   -0.0000    0.0656   -0.0000
   -0.0000    0.0000    0.0000    0.0000   -0.0000    0.0373
```

C

Now we reform our **y** back to images and display:

```
img_a_y = reshape(y(1,:), [256,256]);
img_b_y = reshape(y(2,:), [256,256]);
img_c_y = reshape(y(3,:), [256,256]);
img_d_y = reshape(y(4,:), [256,256]);
img_e_y = reshape(y(5,:), [256,256]);
img_f_y = reshape(y(6,:), [256,256]);

figure
subplot(2,3, 1)
imshow(img_a_y)
subplot(2,3, 2)
imshow(img_b_y)
subplot(2,3, 3)
```

```
imshow(img_c_y)
subplot(2,3, 4)
imshow(img_d_y)
subplot(2,3, 5)
imshow(img_e_y)
subplot(2,3, 6)
imshow(img_f_y)
```