Task 3

I. Feature Detection:

A. Using Harris operator:

function [EFRow,EFCol,Itheta,IMag,I] = GetFeaturesHarris(Im,thresh, Scale)

Function target:

Gets the Feature points in an Image using Harris Operator.

The inputs for the function:

- **Im:** *The Image.*
- **Thresh:** *Value between 0 - 1 where this value is a factor multiplied by the maximum value calculated by Harris operator*.
- **Scale:** *Value to resize the image.*

The outputs of the function:

- **EFRow ,EFCol :** *The Indices of the Feature points*
- **Itheta:** *Image Orientation Matrix*
- **IMag:** *Image Magnitude Matrix.*

B. Using Eigen Values

function [EFRow,EFCol,Itheta,IMag,I] = GetFeaturesEigen(Im,thresh, Scale)

Function target:

Gets the Feature points in an Image using Eigen values.

The inputs of the function:

- **Im:** *The Image.*
- **Thresh:** *Value between 0 - 1 where this value is a factor multiplied by the maximum value of* λ- .
- **Scale:** *Value to resize the image.*

The outputs of the function:

- **EFRow ,EFCol :** *The Indices of the Feature points*
- **Itheta:** *Image Orientation Matri.x*
- **IMag:** *Image Magnitude Matrix.*
- **I:** *Image.*

II. Feature Descriptor:

function [ImageDes] =SIFTDescriptor(EFRow,EFCol,Itheta,IMag,I)

Function target:

Get descriptors for every feature point either detected using Harris operator or Eigen Values saving each descriptor in a 128*1 matrix then saving all descriptors of all features in a big matrix called ImageDes.

The inputs of the function:

- **EFRow ,EFCol :** *The Indices of the Feature points*
- **Itheta:** *Image Orientation Matrix.*
- **IMag:** *Image Magnitude Matrix*
- **I:** *Image.*

The outputs of the function:

ImageDes: *Matrix (128 * number of feature points) Each Column is the Descriptor for a feature point.*

III. Matching using SSD:

function [result]=MatchingHarris(I1,thresh1,Scale1,I2,thresh2, Scale2, SSDthresh)

function [result]=MatchingEigen(I1,thresh1,Scale1,I2,thresh2, Scale2, SSDthresh)

Function target:

This is the main function in the program. It first call the function get features either Harris or Eigen, and record the time it take. Then it call the function SIFTDescriptor(), to get the Descriptor for the feature points and record the time it takes. And repeat the previous steps for the second image. In the end it start matching the two descriptors with each other, decide whether they are matching or not ,record the time matching takes, and shows the two images with feature points on them and the result.

The inputs of the function:

- **I1:** *First Image.*
- **Thresh1:** *Threshold value for the first Image .*
- **Scale1:** *Resize Value for Image 1.*
- **I2:** *Second Image.*
- **Thresh2:** *Threshold value for the second image.*
- **Scale2:** *Resize Value for second image*
- **SSDthresh:** *Maximum value of SSD to say that to feature points match.*

The outputs of the function:

- **Result:** *Matching in case the two images match, Not Matching in case the to images doesn't match.*

Results from our Program

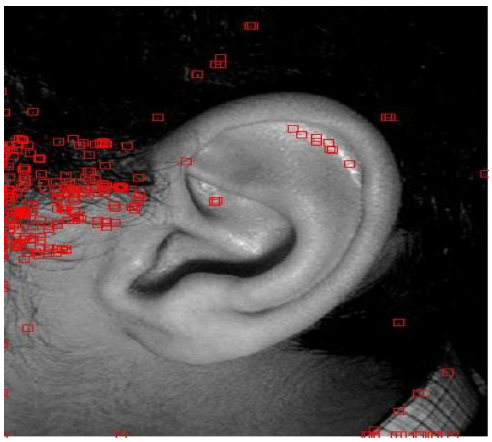
MatchingHarris('MaleEar3.jpg',0.15,1,'MaleEar2.jpg',0.15, 1,0.01)
The original size of similar images :

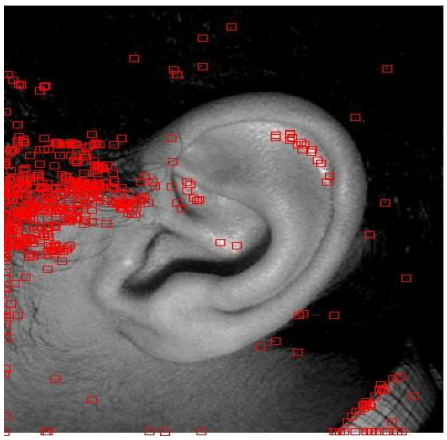
For First Image:

```
The time to get SIFT descriptor for 236 feature points is 0 mins 1.27 secs
      Detecting features using Harris Operator took 0 mins 2.81 secs
```

For Second Image:

```
The time to get SIFT descriptor for 514 feature points is 0 mins 3.001 secs
      Detecting features using Harris Operator took 0 mins 2.73 secs
```





```
Result =  Matching /// Matching Time = 0 mins 0.19 secs
```

Notes

- Since the second image has nearly double the features therefore the SIFT takes double the time in second image but the feature detection time is nearly the same in both because as we calculate the Harris for each point and then make threshold for the whole matrix.
- Matching time depend on the number of feature points in each image.

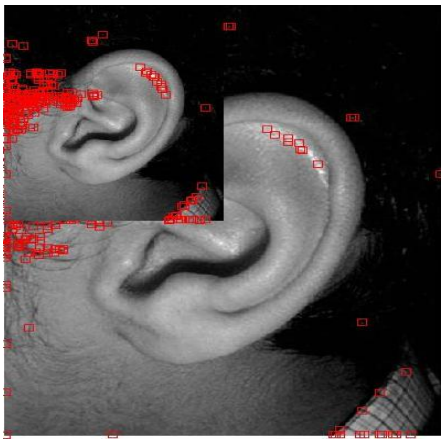
MatchingHarris('MaleEar3.jpg',0.15,0.5,'MaleEar2.jpg',0.15, 0.5,0.01)
The HALF size of similar images :

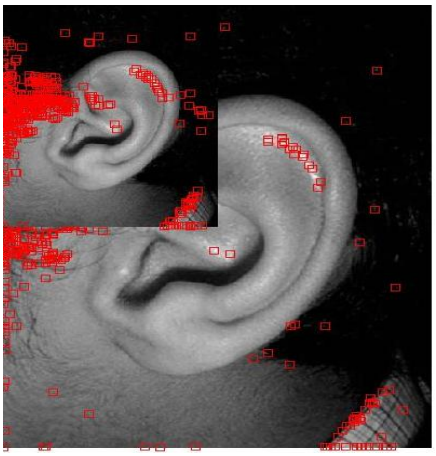
For First Image:

The time to get SIFT descriptor for 217 feature points is 0 mins 1.01 secs
Detecting features using Harris Operator took 0 mins 0.76 secs

For Second image:

The time to get SIFT descriptor for 361 feature points is 0 mins 1.84 secs
Detecting features using Harris Operator took 0 mins 0.82 secs





Result = Matching /// Matching Time = 0 mins 0.14 secs

Notes

- Harris Operator took less time as the number of pixels is less and it is calculated to each pixel.
- Number of Feature points decreased as size of the image decreased (less details).
- SIFT Descriptor time also decreased but not that much as it is dependent only on the number of features and not the size of the image.
- Matching occurred with same scales and the same with quarter size of two images

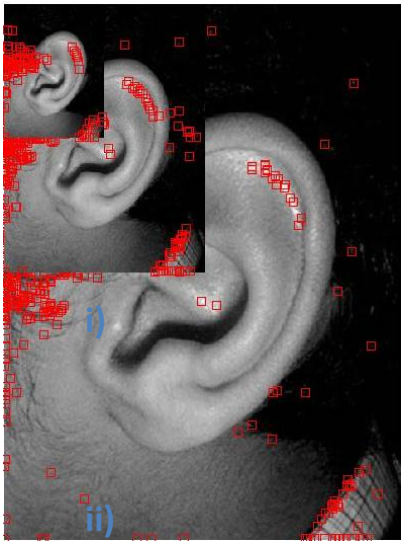
MatchingHarris('MaleEar3.jpg',0.15,0.25,'MaleEar2.jpg',0.15, 0.25,0.01)
The Quarter size of similar images :

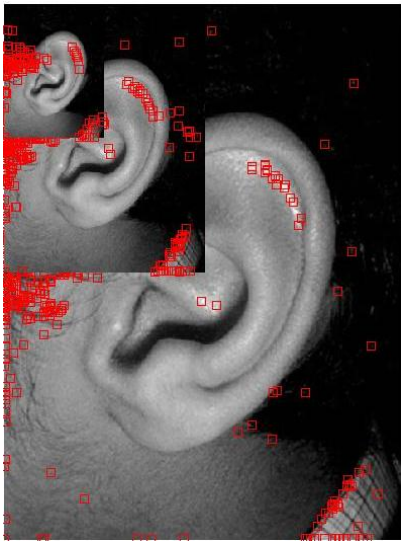
For First Image:

```
The time to get SIFT descriptor for 45 feature points is 0 mins 0.16 secs
    Detecting features using Harris Operator took 0 mins 0.24 secs
```

For Second Image:

```
The time to get SIFT descriptor for 82 feature points is 0 mins 0.3 secs
    Detecting features using Harris Operator took 0 mins 0.3 secs
```





```
Result =  Matching /// Matching Time = 0 mins 0.01 secs
```

Notes

- Features points decreased and time also decreased more.

MatchingHarris('MaleEar1.jpg',0.15,1, '2C1.jpg',0.05, 1, 0.01)

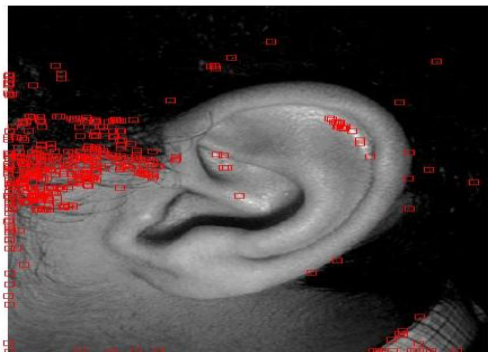
And for different images:

First image:

The time to get SIFT descriptor for 16 feature points is 0 mins 0.244 secs
Detecting features using Harris Operator took 0 mins 5.072 secs

Second image:

The time to get SIFT descriptor for 424 feature points is 0 mins 5.163 secs
Detecting features using Harris Operator took 0 mins 6.282 secs



Result = Not Matching //// Matching Time = 0 mins 0.021 secs

Notes

- Two different images didn't match.

Appling the same for Matching Eigen.

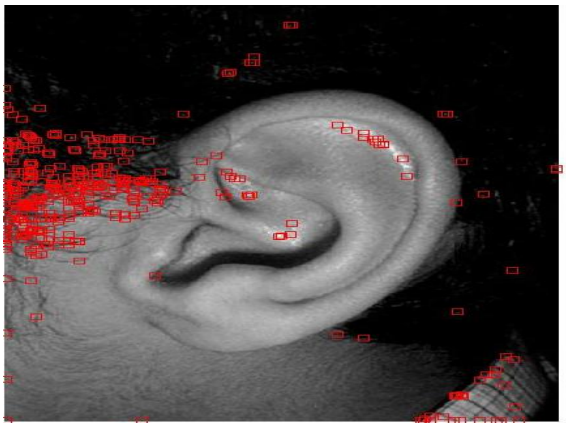
MatchingEigen('MaleEar3.jpg',0.15, 1, 'MaleEar2.jpg',0.15, 1, 0.01)

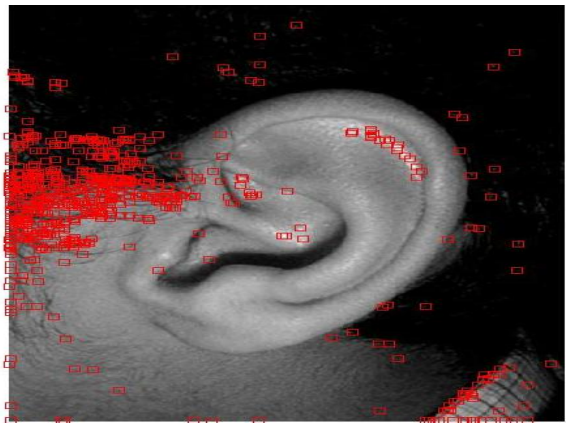
First image:

The time to get SIFT descriptor for 372 feature points is 0 mins 4.509 secs
Detecting features using Eigen Values took 0 mins 3.213 secs

Second Image:

The time to get SIFT descriptor for 637 feature points is 0 mins 9.081 secs
Detecting features using Eigen Values took 0 mins 3.293 secs





Result = Matching /// Matching Time = 0 mins 0.686 secs

Notes

- The timing using Eigen minimum is larger so Eigen algorithm takes more time than Harris and more features are extracted

MatchingEigen('MaleEar3.jpg',0.15, 0.5, 'MaleEar2.jpg',0.15, 0.5, 0.01)

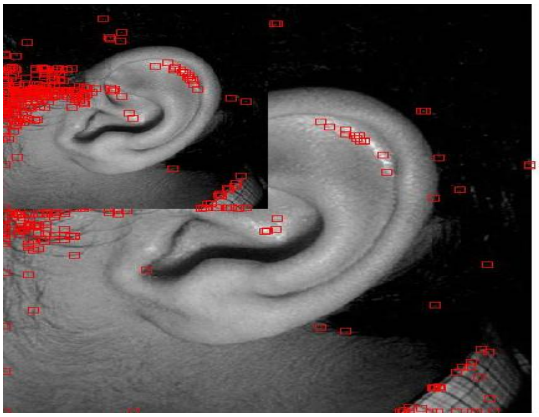
Half scale of two images:

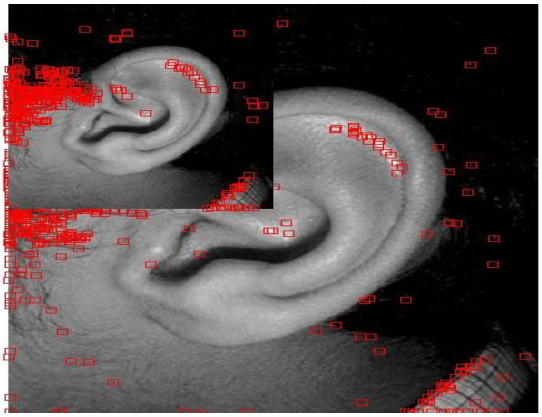
First Image:

The time to get SIFT descriptor for 247 feature points is 0 mins 2.449 secs
Detecting features using Eigen Values took 0 mins 0.811 secs

Second Image:

The time to get SIFT descriptor for 273 feature points is 0 mins 2.496 secs
Detecting features using Eigen Values took 0 mins 0.826 secs





Result = Matching //// Matching Time = 0 mins 0.203 secs

MatchingEigen('MaleEar3.jpg',0.15, 0.25, 'MaleEar2.jpg',0.15, 0.25, 0.01)

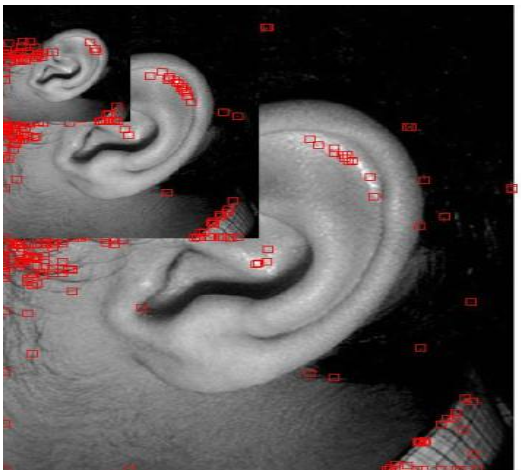
Quarter Scale of two images:

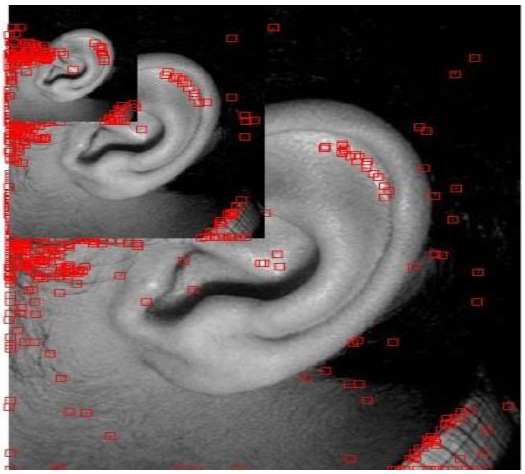
First Image:

The time to get SIFT descriptor for 43 feature points is 0 mins 0.344 secs
Detecting features using Eigen Values took 0 mins 0.218 secs

Second Image:

The time to get SIFT descriptor for 96 feature points is 0 mins 0.687 secs
Detecting features using Eigen Values took 0 mins 0.327 secs





Result = Matching /// Matching Time = 0 mins 0 secs

MatchingEigen('MaleEar1.jpg',0.15,1, '2C1.jpg',0.1, 1, 0.01)

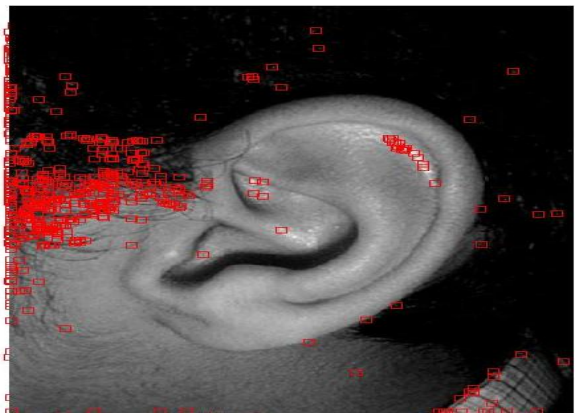
Different images:

First Image:

The time to get SIFT descriptor for 11 feature points is 0 mins 0.156 secs
Detecting features using Eigen Values took 0 mins 3.136 secs

Second Image:

The time to get SIFT descriptor for 439 feature points is 0 mins 5.32 secs
Detecting features using Eigen Values took 0 mins 5.257 secs



Result = Not Matching /// Matching Time = 0 mins 0.015 secs