03.face_space

April 21, 2018

```
In [35]: import numpy as np
         from eigenface.yalefaces import YaleFaceDb
         from eigenface.eigenfaces import calculate_eigen_faces
         from eigenface.eigenfaces import gen_images
         np.set_printoptions(precision=2, suppress=True, formatter={'float': '{: 0.2f}'.format
```

```
In [36]: db = YaleFaceDb()
         images = gen_images()
         # images = db.get_list().astype(dtype=np.float)
         labels = db.get_label()
```

```
In [37]: (mean_images, eigen_faces), (vector_mean_matrix, mean_vector, eigen_value, norm_ui) =
```

```
In [159]: def get_weight(face_vector, mean_vector, norm_ui, size):
              theta = face_vector - mean_vector
              return np.matmul(norm_ui[:size], theta)

          def find_size(eigen_value, percent = 0.9):
              total = eigen_value.sum()
              for i in range(len(eigen_value)):
                  size = i + 1
                  cur  = eigen_value[:size].sum()
                  if cur/float(total)>=percent:
                      return size
              return len(eigen_value)

          def get_all_weight(images, mean_vector, norm_ui, size):
              w = [get_weight(images[i,:,:].flatten(), mean_vector, norm_ui, size)for i in rang
              return w

          def distance_classify(w, weights):
              diff = weights - w
              norms = np.linalg.norm(diff, axis=1)
              closest_face_id = np.argmin(norms)
              return closest_face_id

          size = find_size(eigen_value)
          w = get_all_weight(images, mean_vector, norm_ui, size)
          print(w)
```

```
[array([-0.43,  1.43]), array([-2.01, -0.92]), array([ 2.44, -0.51])]
```

```
In [161]: x = np.array([2.44, -0.51])
          print(distance_classify(x, w))
```

```
2
```