Principal Component Analysis

(Programming Assignment 1)

- By Group 8

170030019 Abhinav Gupta

170020009 Rijul Kumar

GITHUB LINK FOR CODE :

https://github.com/Abhi10398/Machine-Learning/blob/main/Dimensionality%20Reduction/PCA/PCA.ipynb

PCA Steps

1. Read the Original Images in data form.
2. Flatten the images and append the data of images in column of a (4096,30) matrix.
3. Normalize the (4096,30) matrix.
4. Calculating the covariance. (4096 x 4096)
5. Calculating EigenValues(4096) and EigenVectors (4096 x 4096) of covariance matrix and ordering them in decreasing order of EigenValues.
6. Compress the data by projecting the original data over first N eigenvector (Principal components) (N x 30)
7. Using compressed data (N x 30) , reconstruct the images by projecting it to Original 4096 axis, resulting in reconstructed image (4096 x 30)
8. Plot the reconstructed images after reshaping it and compare it with the original one.

Normalizing and denormalizing

To normalize the dataset we took the mean and standard deviation and then normalize the data using :

Data_norm = (data - mean)/std

To denormalize the data we first reconstruct the image from the compressed image using projection of N x 30 images to 4096 components resulting in reconstructed image (4096 x 30)

Recon_image_denorm = (Recon_image * std) + mean

Code Snippet 1

Calculates the covariance matrix and then its eigenvalues and eigenvectors

```
np.set_printoptions(precision=3) # to limit the calculations
cov = np.cov(X_norm) # create a covariance matrix
print("covariance shape : ",cov.shape)

# Eigen Values
EigVal,EigVec = np.linalg.eig(cov)  # Find eigen value and eigen vector of covariance matrix

print("Eigenvalues size:", EigVal.shape,"\n")
```

```
covariance shape :  (4096, 4096)
Eigenvalues size: (4096,)
```

Code Snippet 2

Arrange the eigenvectors in descending order

```
# Ordering Eigen values and vectors
# arrange the eigen values in descending order
order = EigVal.argsort()[::-1]
EigVal = EigVal[order]
EigVec = EigVec[:,order]

#Projecting data on Eigen vector directions resulting to Principal Components
EigVec=EigVec.astype(float)
PC = (X_norm.T @ EigVec).T   #cross product
```

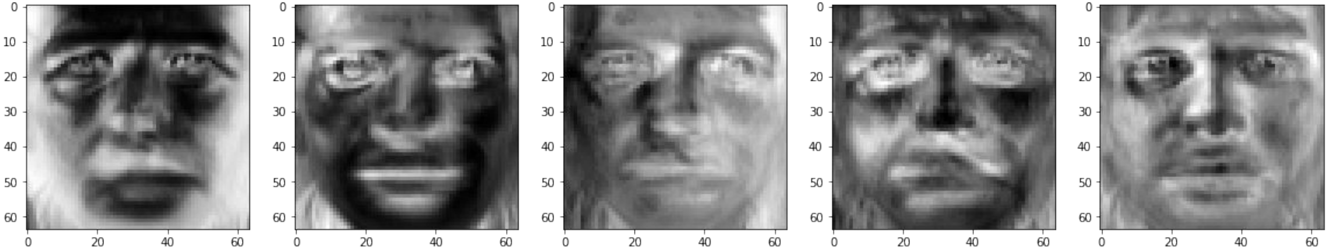
Code Snippet 3

Compress the data and then reconstruct image using it, followed by denormalization

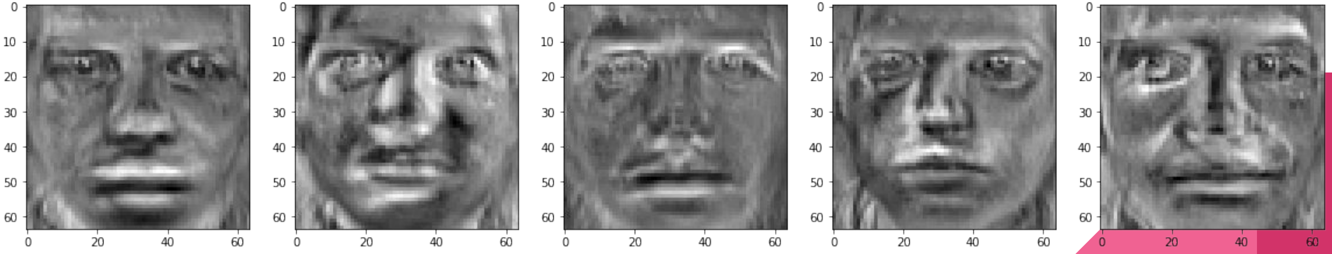
```
def reconstruct(X_norm,EigVec,EigVal,mean,std,n):  # function to compress and then reconstruct
  # Compress the image by taking projection of actual data on first n PCs
  PC = (X_norm.T @ EigVec[:,:n]).T   #cross product
  # PC shape = n x 30

  # Recover the image by taking projection of actual data by cross multiplication of EigVec.T
  recover = (EigVec[:,:n] @ PC)*std + mean  # recover shape = 4096 x 30
  return recover
```

Please refer the complete code.

Visualization of Principal Components

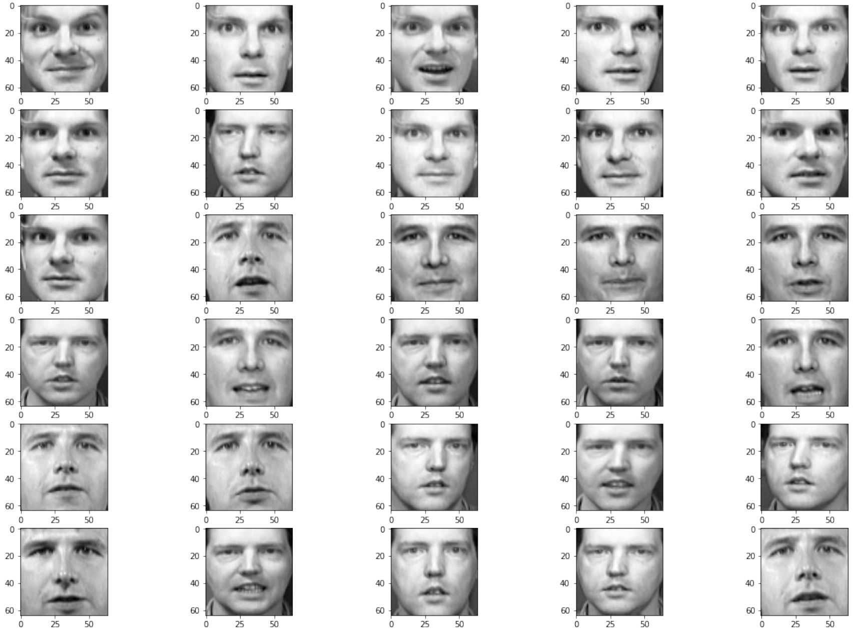


Here are the top 10 eigenvectors, This shows exactly what features our algorithm is trying to extract.

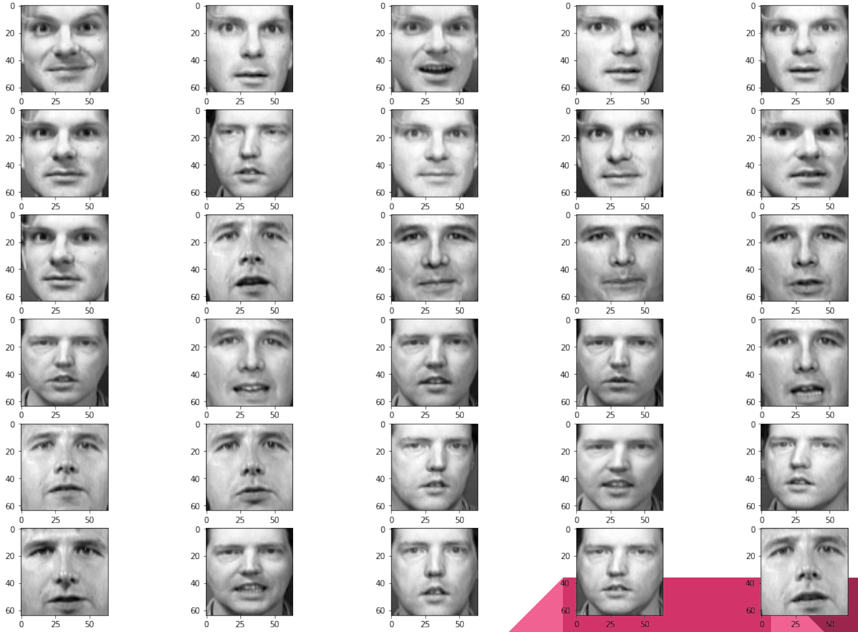


Reconstructing Original Images

Original Images

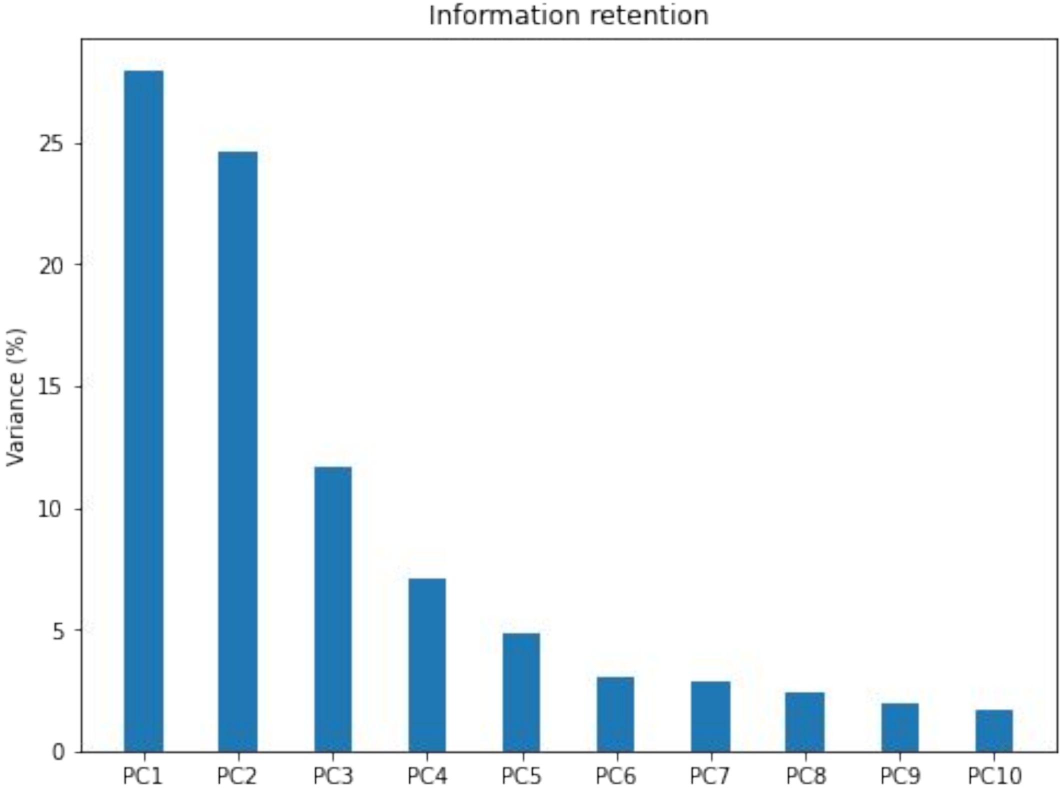


Reconstructed Images using 4096 comp



Because we compressed to 4096 components by projection and then reconstructed it , that means technically no compression and no data loss as original image is also of 4096 components. This can be seen it above images too.

Information retained by PC



This is the amount of information retained for first 10 eigenvectors i.e principal components

They are basically the eigenvalue percentage only

It also shows that first 3 to 5 PCs are mostly responsible for more than 90% of data

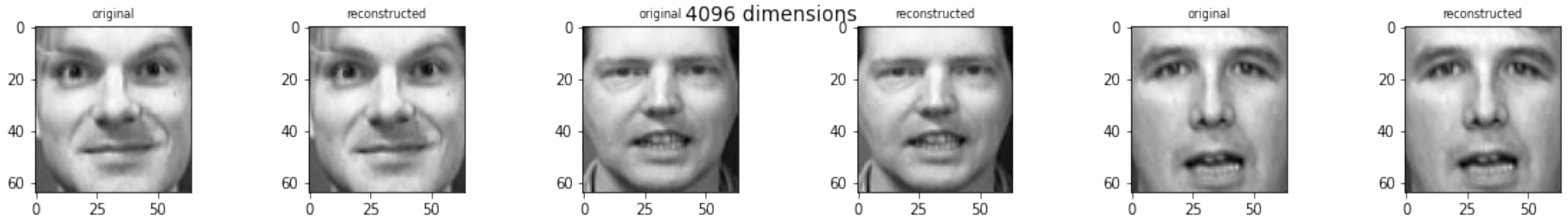
Lets visualize the reconstructed images for various compressions by taking different number of principal components

Note :
Original image = (4096 x 1) i.e (64 x 64)
compressed image = (N x 1)
Reconstructed image = (4096 x 1) i.e (64 x 64)

Comparison between all 3 person’s original and reconstructed images

Taking First 4096 Principal Components

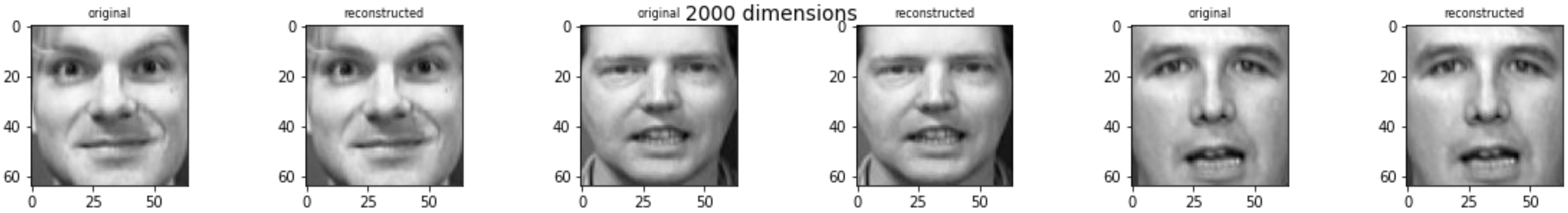
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person’s original and reconstructed images

Taking First 2000 Principal Components

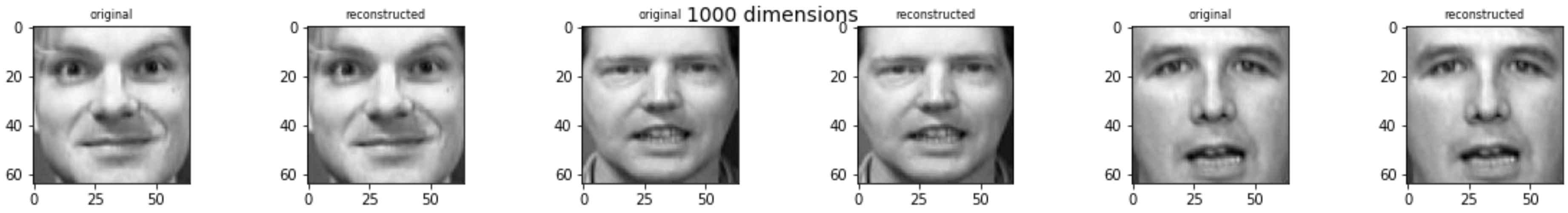
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person's original and reconstructed images

Taking First 1000 Principal Components

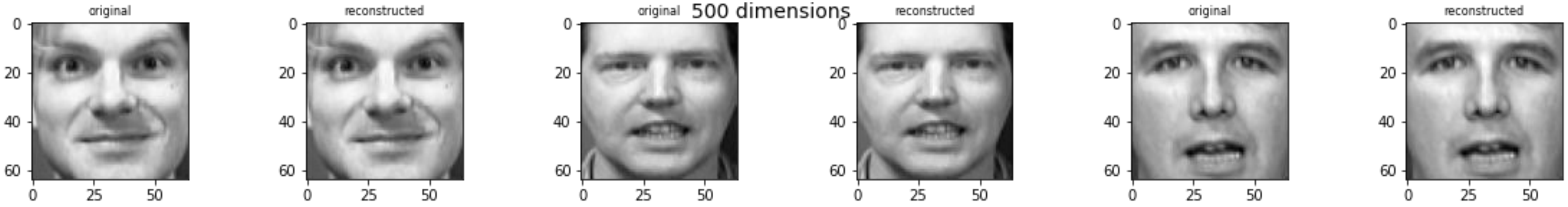
Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Comparison between all 3 person’s original and reconstructed images

Taking First 500 Principal Components

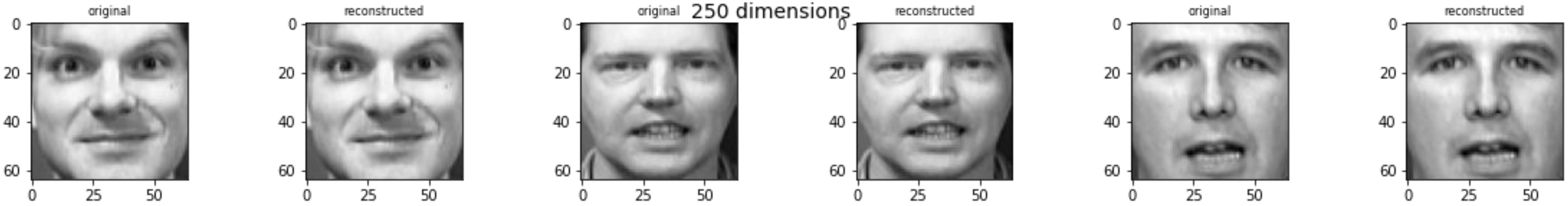
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person's original and reconstructed images

Taking First 250 Principal Components

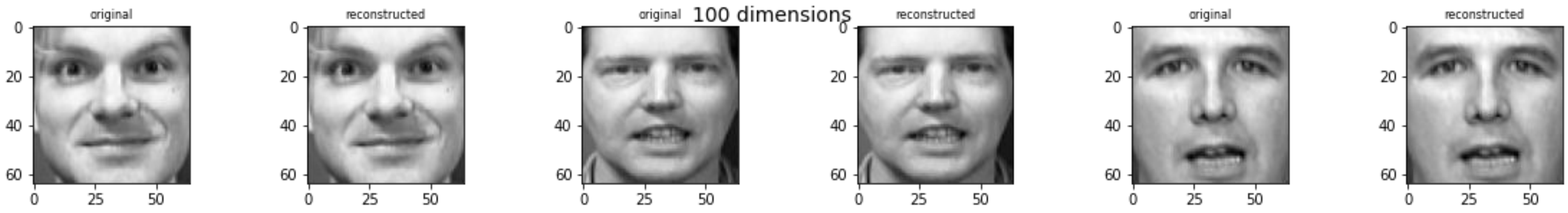
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person’s original and reconstructed images

Taking First 100 Principal Components

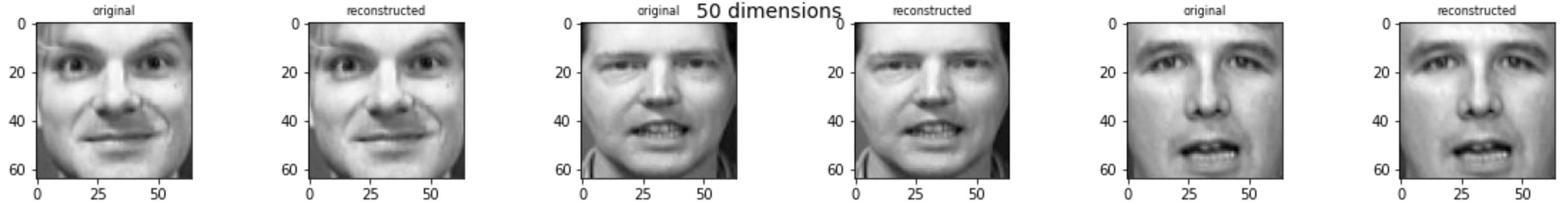
Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Comparison between all 3 person’s original and reconstructed images

Taking First 50 Principal Components

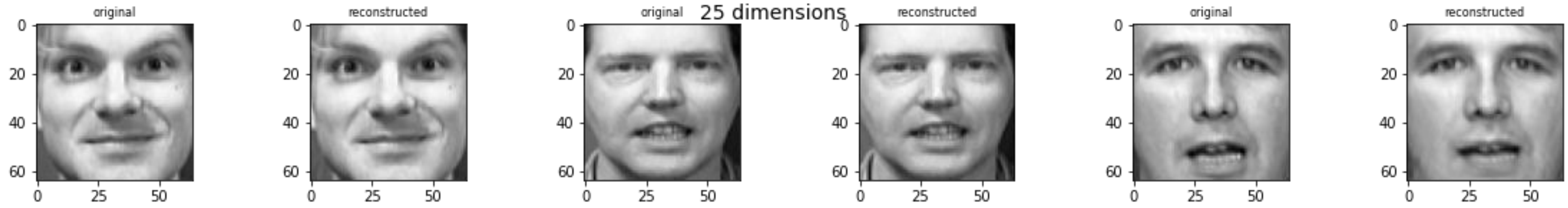
Original **Reconstructed** **Original** **Reconstructed** **Original** **Reconstructed**



Comparison between all 3 person’s original and reconstructed images

Taking First 25 Principal Components

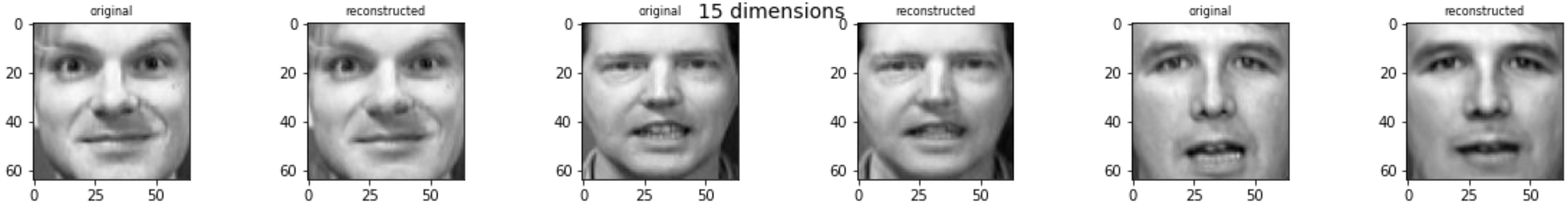
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person’s original and reconstructed images

Taking First 15 Principal Components

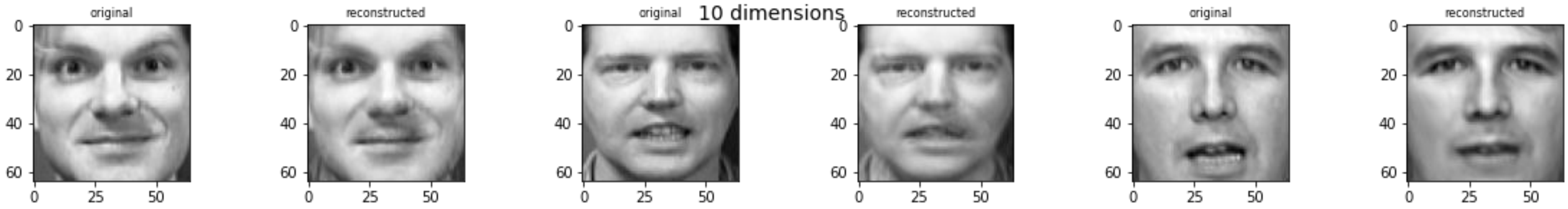
Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Comparison between all 3 person’s original and reconstructed images

Taking First 10 Principal Components

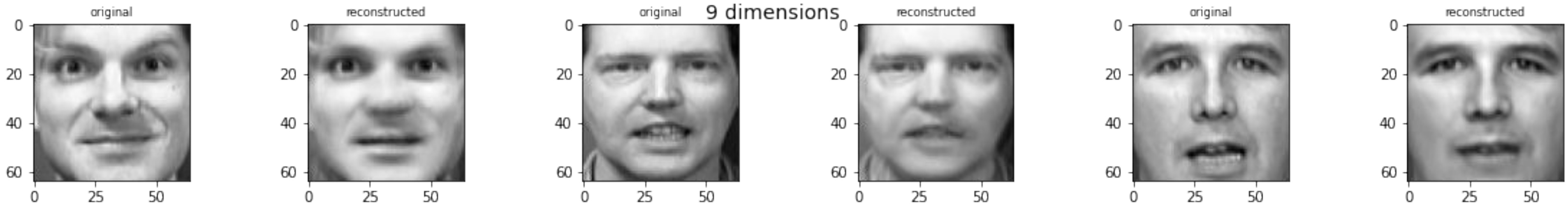
Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Comparison between all 3 person’s original and reconstructed images

Taking First 9 Principal Components

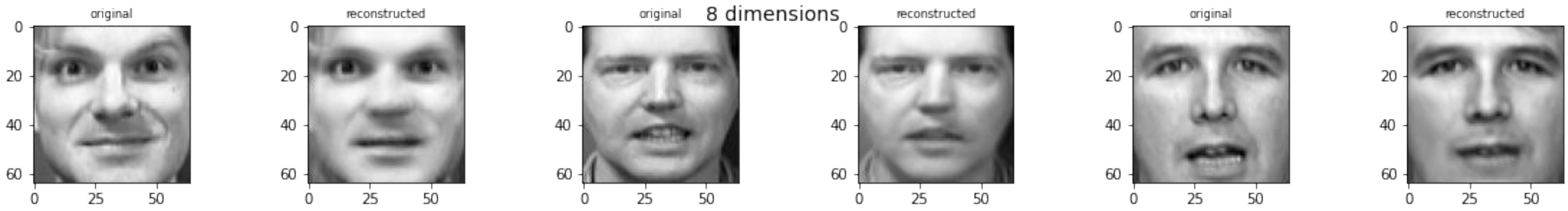
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person’s original and reconstructed images

Taking First 8 Principal Components

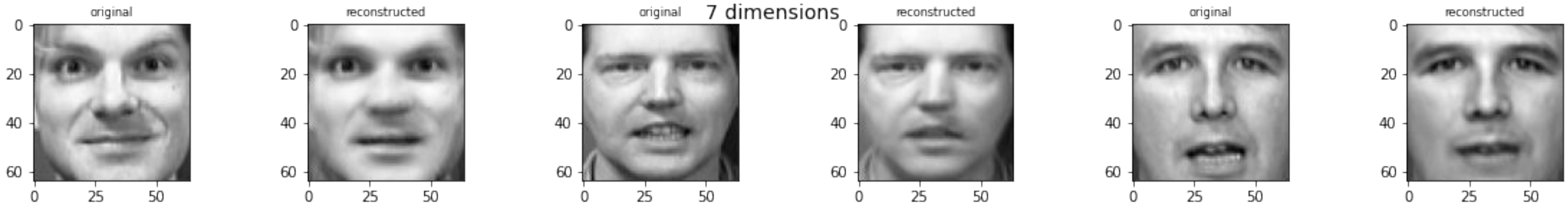
Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Comparison between all 3 person's original and reconstructed images

Taking First 7 Principal Components

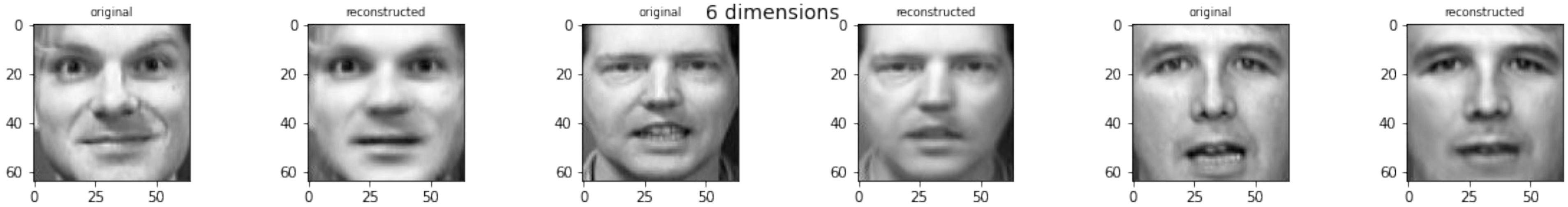
Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Comparison between all 3 person's original and reconstructed images

Taking First 6 Principal Components

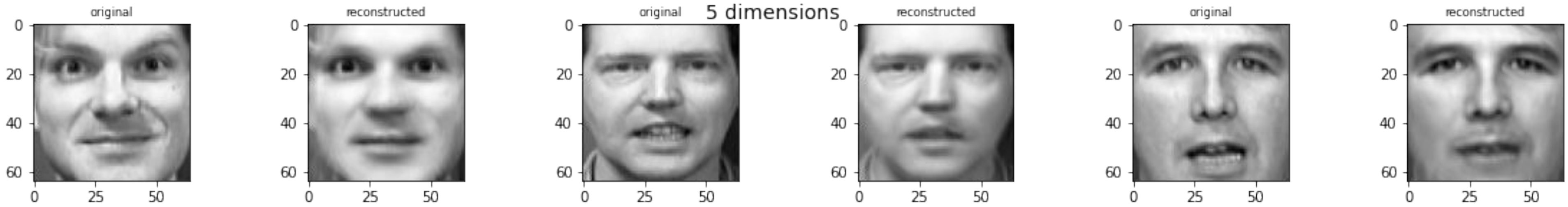
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person’s original and reconstructed images

Taking First 5 Principal Components

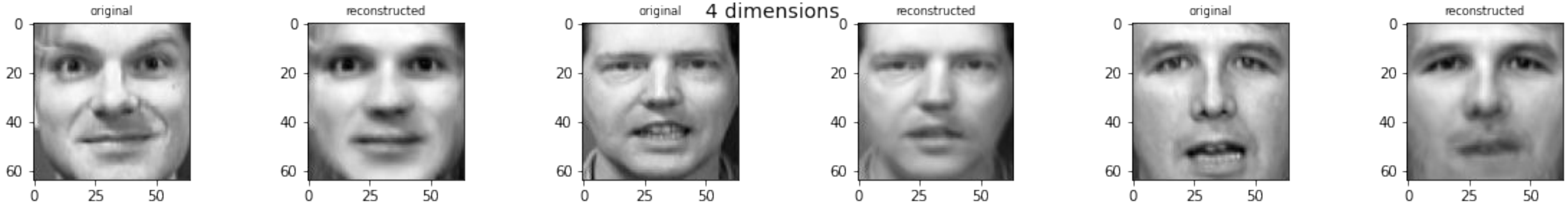
Original **Reconstructed** **Original** **Reconstructed** **Original** **Reconstructed**



Comparison between all 3 person’s original and reconstructed images

Taking First 4 Principal Components

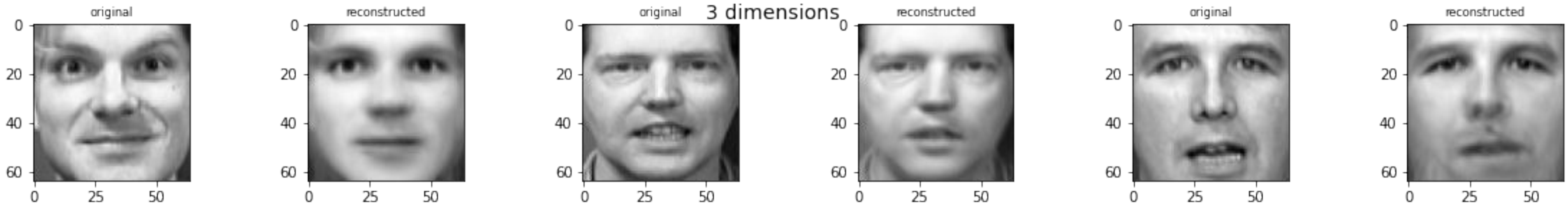
Original **Reconstructed** **Original** **Reconstructed** **Original** **Reconstructed**



Comparison between all 3 person's original and reconstructed images

Taking First 3 Principal Components

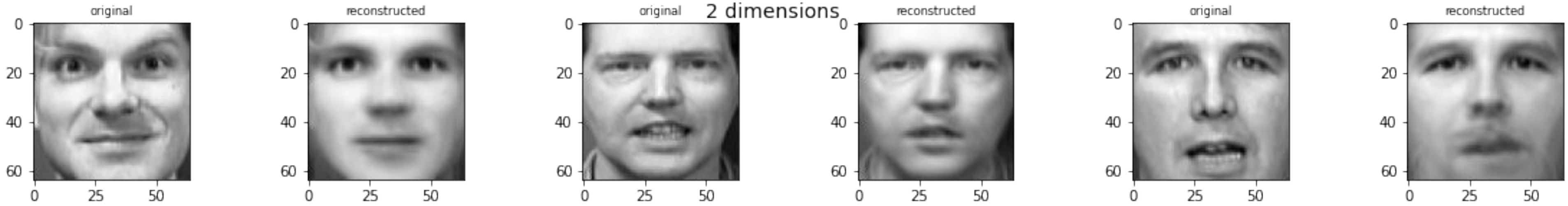
Original Reconstructed Original Reconstructed Original Reconstructed



Comparison between all 3 person’s original and reconstructed images

Taking First 2 Principal Components

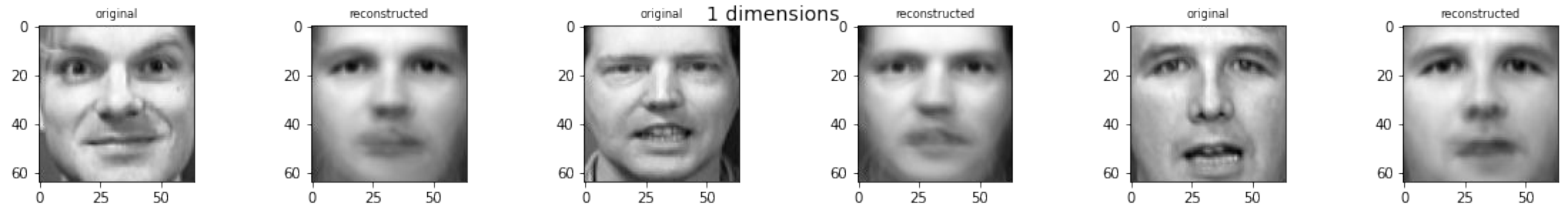
Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Comparison between all 3 person’s original and reconstructed images

Taking First 1 Principal Component

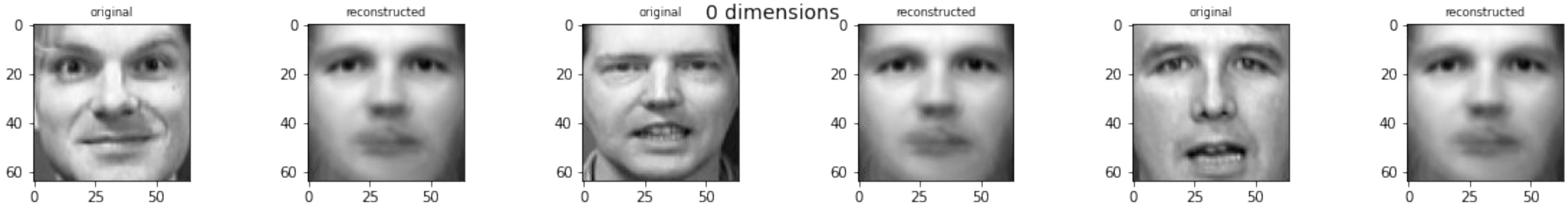
Original | Reconstructed | Original | Reconstructed | Original | Reconstructed



Comparison between all 3 person’s original and reconstructed images

Taking First 0 Principal Components

Original | **Reconstructed** | **Original** | **Reconstructed** | **Original** | **Reconstructed**



Observations regarding different PC

- For person 1, lips and lower jaw of the face begins becoming hazy at 10 principal components.
- For person 2, teeth, lips and lower jaw starts becoming hazy at 10 principal components.
- For person 3, teeth and lips starts becoming hazy at 10 principal components.
- So for 10 PCs we could hardly discriminate the minor features of face like mouth, teeth, eyes etc.
- 15 is the most optimized value till which we can retain almost whole data without much data loss.

Observations regarding different PC

- Till 5 PCs we are able to discriminate among the images of 3 people. Although facial features are not clearly visible but still ,visible enough to discriminate among the 3.
- Till **1 PCs (value of x)**, it looks like a face, after it we are unable to perceive images as faces with proper facial features.
- All images of 3 people becomes roughly same at 1 principal component.
- All images of 3 people becomes same at 0 principal components because of denormalization.

THE END