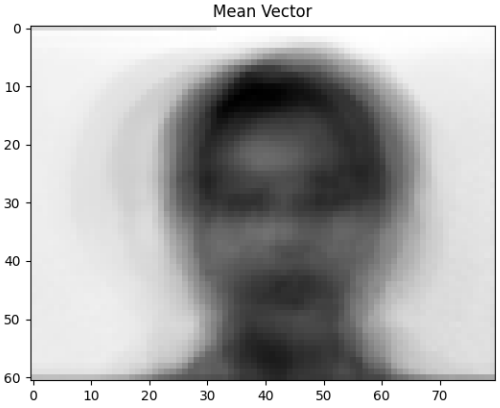
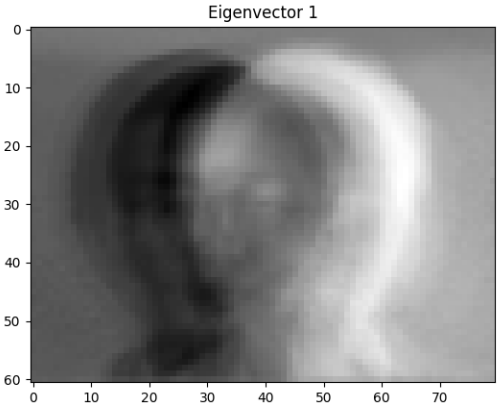
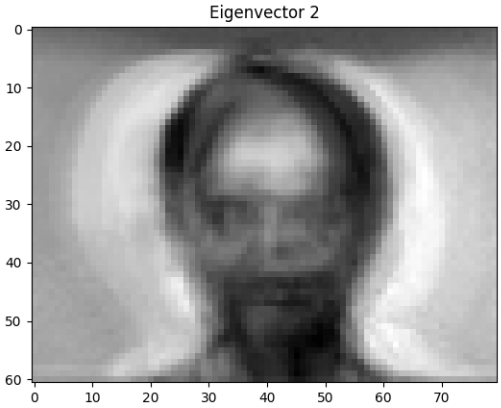
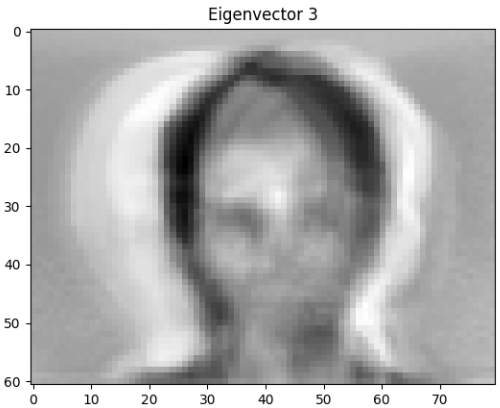
郭雅美 B09902085 Programming Report

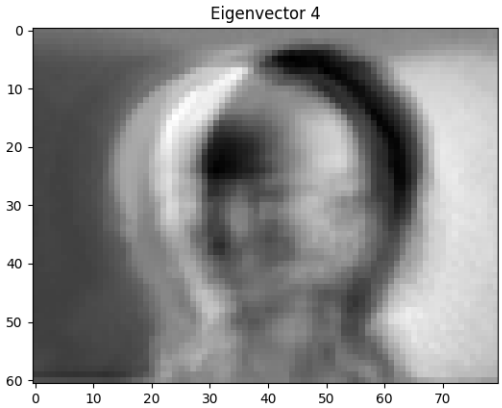
a)



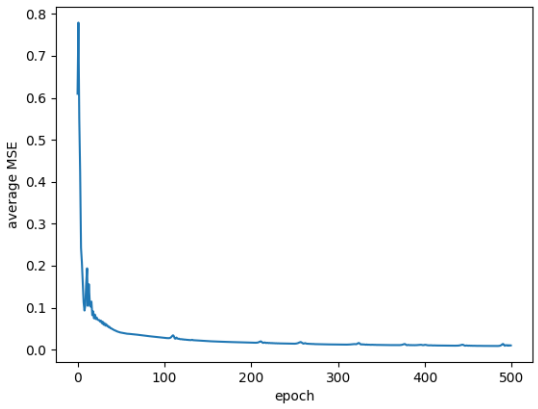




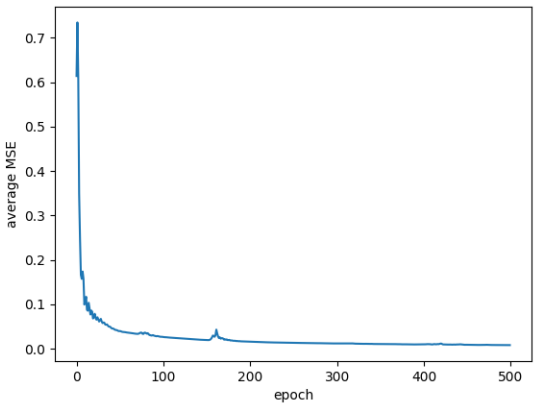




b) Training curve of Autoenconder



Training Curve of DenoisingAutoenconder



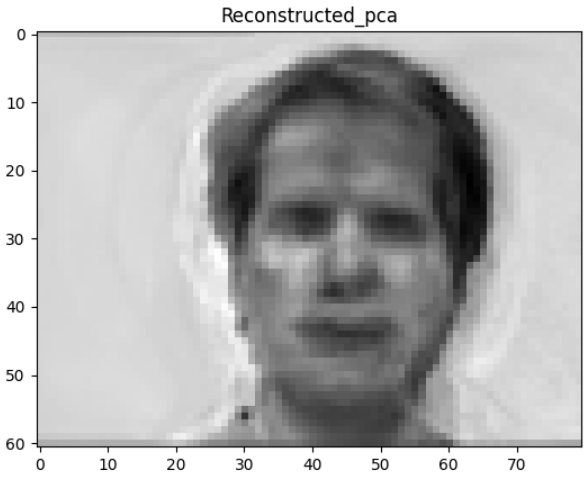
c)

The full result:

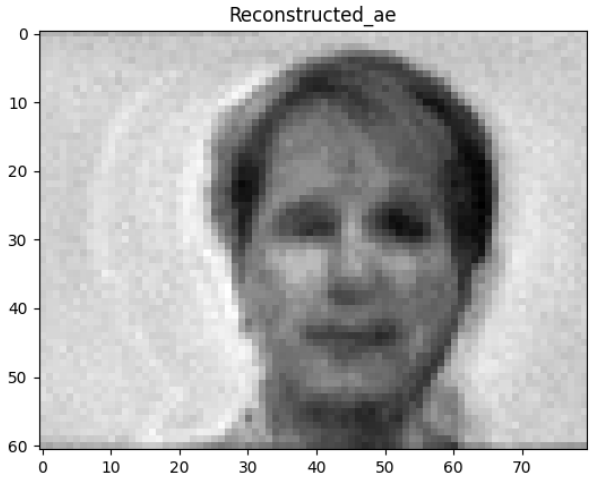
```
(fai_hw4) alfali@Alfalis-MacBook-Pro hw4 % python3 hw4.py
Loading data...
PCA Training Start...
Autoencoder Training Start...
DenoisingAutoencoder Training Start...
Feature Transformation
Logistic Regression Training Start...
Acc from PCA: 0.9333333333333333
Acc from Autoencoder: 0.9333333333333333
Acc from DenoisingAutoencoder: 0.9333333333333333
Reconstruction Loss with PCA: 0.010710469688056324
Reconstruction Loss with Autoencoder: 0.013945752372556209
Reconstruction Loss with DenoisingAutoencoder: 0.013145690081229114
```

Comparison is on the next page.

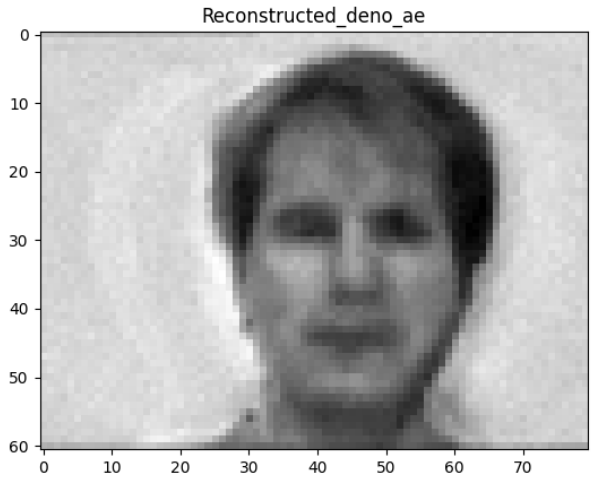
Original image and PCA
MSE PCA: 0.0107



Original image and Autoenconder
MSE Autoenconder: 0.0139



Original image and DenoisingAutoenconder
MSE DenoisingAutoenconder: 0.0131



d)

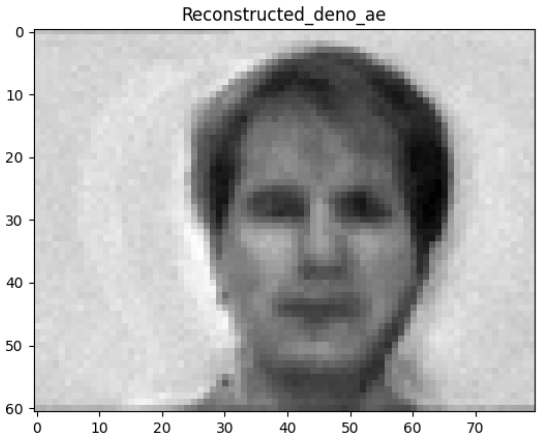
Architecture 1:

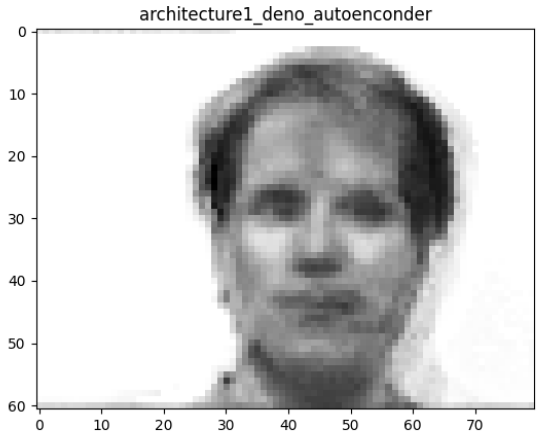
```
self.encoder = nn.Sequential(
    nn.Linear(input_dim, 2048),
    nn.ReLU(),
    nn.Linear(2048, 1024),
    nn.ReLU(),
    nn.Linear(1024, 512),
    nn.ReLU(),
    nn.Linear(512, encoding_dim//2),
    nn.ReLU()
)
self.decoder = nn.Sequential(
    nn.Linear(encoding_dim//2, 512),
    nn.ReLU(),
    nn.Linear(512, 1024),
    nn.ReLU(),
    nn.Linear(1024, 2048),
    nn.ReLU(),
    nn.Linear(2048, input_dim ),
    nn.Sigmoid()
)
```

```
Acc from DenoisingAutoencoder: 0.8
Reconstruction Loss with DenoisingAutoencoder: 0.013035096977622508
```

Left image: Denoising Autoencoder with original architecture.

Right image: Denoising Autoenconder with the architecture above (Architecture 1)





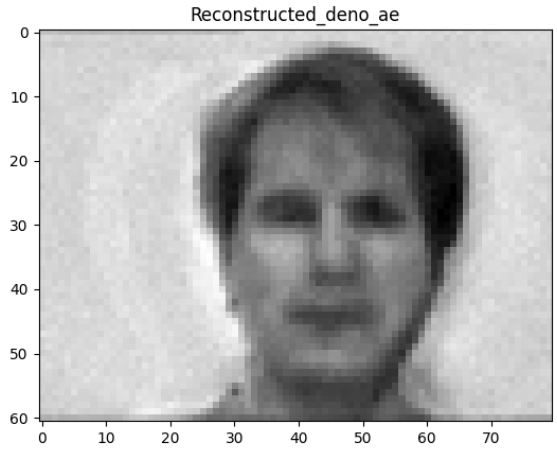
I was expecting the deeper architecture to have clearer and have more visible features than the original architecture, but instead, architecture 1 seems to be blurrier, but nonetheless, their MSE is similar. (architecture 1 MSE: 0.0130, original architecture MSE: 0.0131). Perhaps some information was lost due to increasing number of layers. This architecture has lower accuracy than the original one.

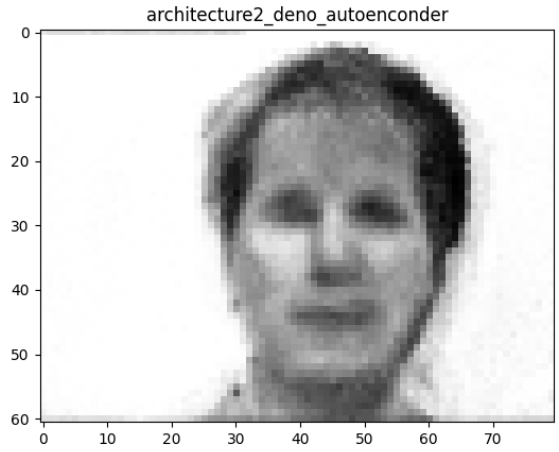
Architecture 2

```
self.encoder = nn.Sequential(
    nn.Linear(input_dim, encoding_dim//2),
    nn.ReLU()
)
self.decoder = nn.Sequential(
    nn.Linear(encoding_dim//2, input_dim),
    nn.Sigmoid()
)
```

```
Acc from DenoisingAutoencoder: 0.5
Reconstruction Loss with DenoisingAutoencoder: 0.012141240281579571
```

Left image: Denoising Autoencoder with original architecture.
Right image: Denoising Autoenconder with the architecture above (Architecture 2)





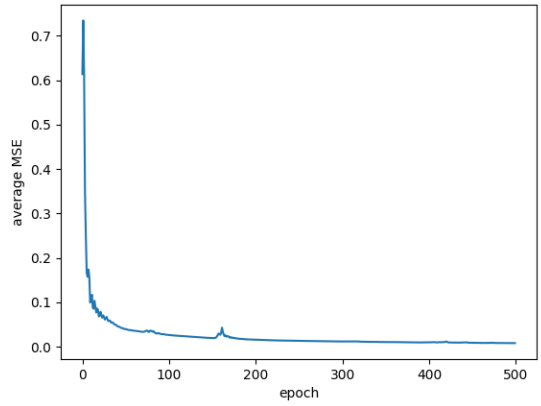
The image reconstructed using architecture 2 seems pixelated, but the MSE is lower than the original architecture (0.0121 < 0.0131). However, the accuracy is just 0.5. I think underfitting might have occur since it only has 2 layers, which somehow made the MSE to reduce for this particular dataset.

e)

<u>All of these used DenoisingAutoenconder</u>

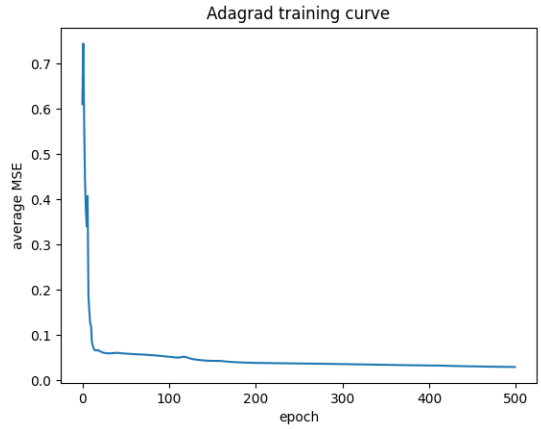
Optimizer using Adam (the original one)

Accuracy = 0.93



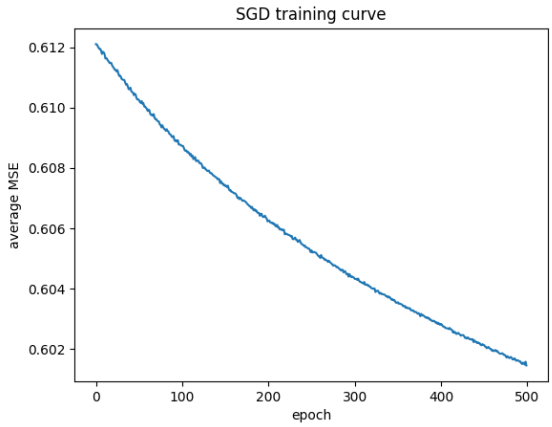
Optimizer using Adagrad

Accuracy: 0.8



Optimizer using SGD (Stochastic Gradient Descent)

Accuracy: 0.9



All of them have a learning rate of 0.001.

Adagrad has a much faster convergence speed when compared to Adam (less flunctuation). However, Adam (acc: 0.93) has a better overall performance when compared to Adagrad (acc: 0.8)

SGD has a lower convergence speed when compared to Adam (the slope of the graph for SGD is not steep). SGD perform slightly worse (acc: 0.9) when compared to Adam (acc: 0.93, consistent between 0.9 – 0.93)