

#1

will compute the same way we discussed in class.

a) $P(\text{Water} = \text{cool} | \text{Play} = \text{yes}) = \boxed{\frac{1}{3}}$

$$P(\text{Water} = \text{cool} | \text{Play} = \text{no}) = \boxed{0}$$

b) $P(\text{Play} = \text{yes} | \text{Water} = \text{warm}) = \boxed{\frac{2}{3}}$

$$P(\text{Play} = \text{no} | \text{Water} = \text{warm}) = \boxed{\frac{1}{3}}$$

c) $P(\text{Play} = \text{yes} | \text{Humid} = \text{high}) = \boxed{\frac{2}{3}}$

$$P(\text{Play} = \text{yes} | \text{Humid} = \text{normal}) = \boxed{1}$$

d) $P(\text{Water} = \text{cool} | \text{Play} = \text{yes}) = \frac{1+2}{3+2} = \boxed{\frac{2}{5}}$

$$P(\text{Water} = \text{cool} | \text{Play} = \text{no}) = \frac{0+1}{1+2} = \boxed{\frac{1}{3}}$$

Also as a side check, we discussed the following in class:

$$P(A|B) + P(\bar{A}|B) = 1 \rightarrow \text{always true}$$

$$P(A|B) + P(A|\bar{B}) = 1 \rightarrow \text{not always true.}$$

→ this is the pattern we can observe above, for example, in b), we have $\frac{2}{3} + \frac{1}{3} = 1$, but $\neq 1$ everywhere else.

QUESTION 2

ANSWER

a) $K(x, z) = k_1(x, z) k_2(z, z)$

The product of these matrices will give a positive semi-definite matrix, since the product of two positive semi-definite matrices is itself a positive semi-definite matrix.

- b) will not be a valid kernel.
↳ different from the input scaling property.
- c) will be a valid kernel.

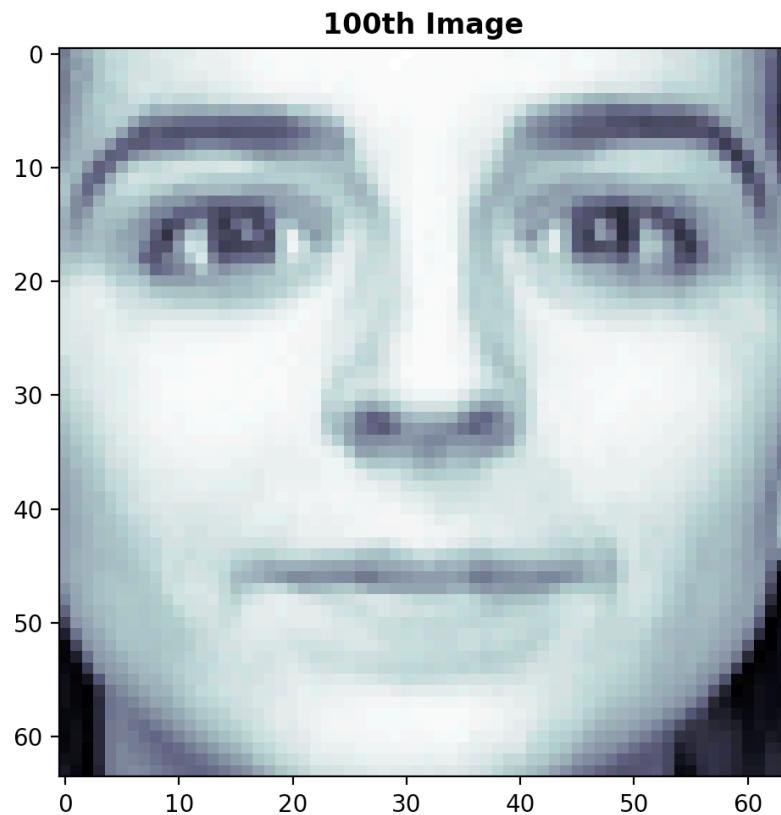
Note: I have miscalculated the time and spent too long on question 3, and was not able to show work for this question.

↳ I know that normally we can use 3 different methods to show whether a kernel is a valid kernel.

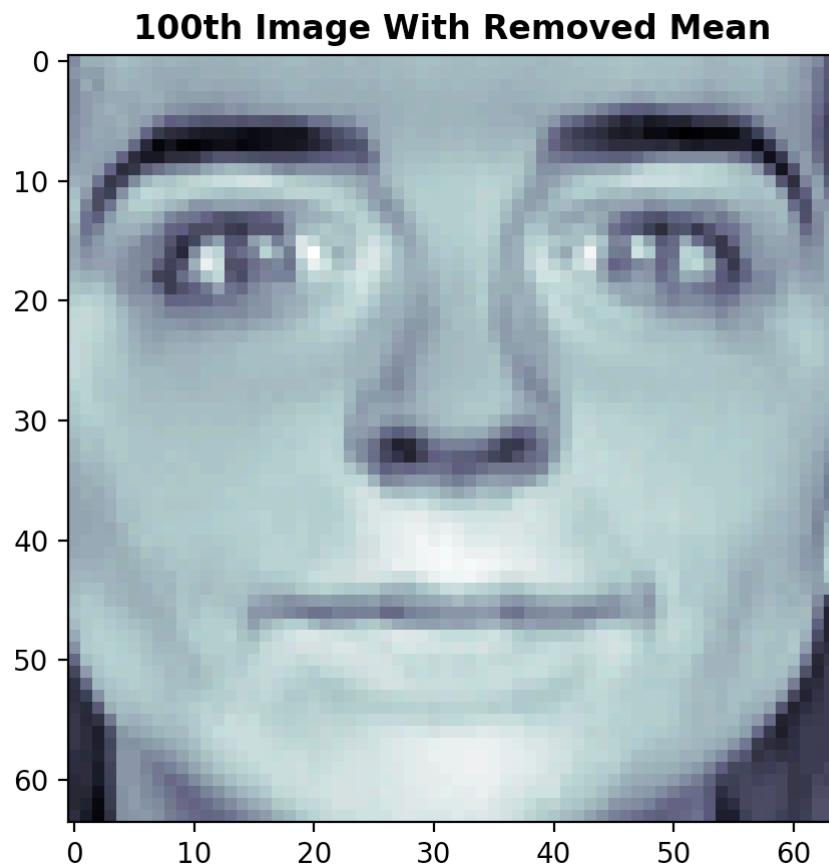
ANSWER

Question 3.

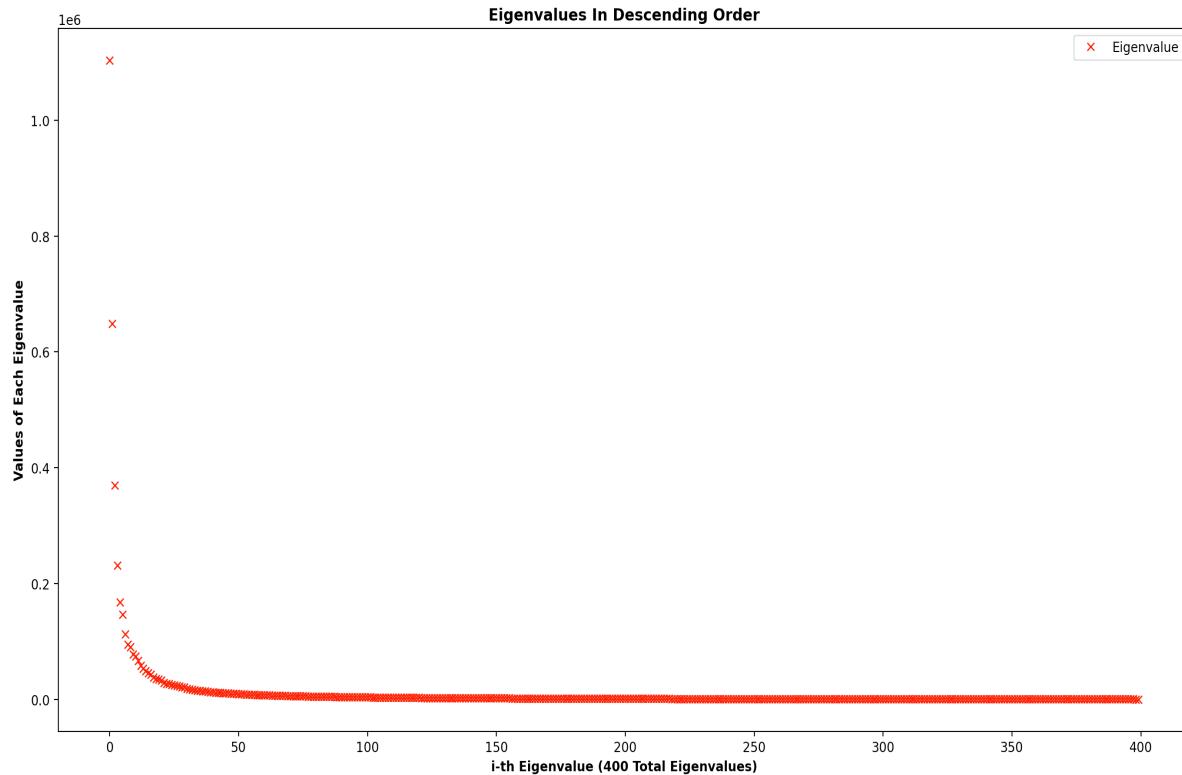
a) Here is the 100th image:



b) Here is the 100th image after the mean of the images was removed:



c) Here is the plot of the eigenvalues sorted in the descending order:

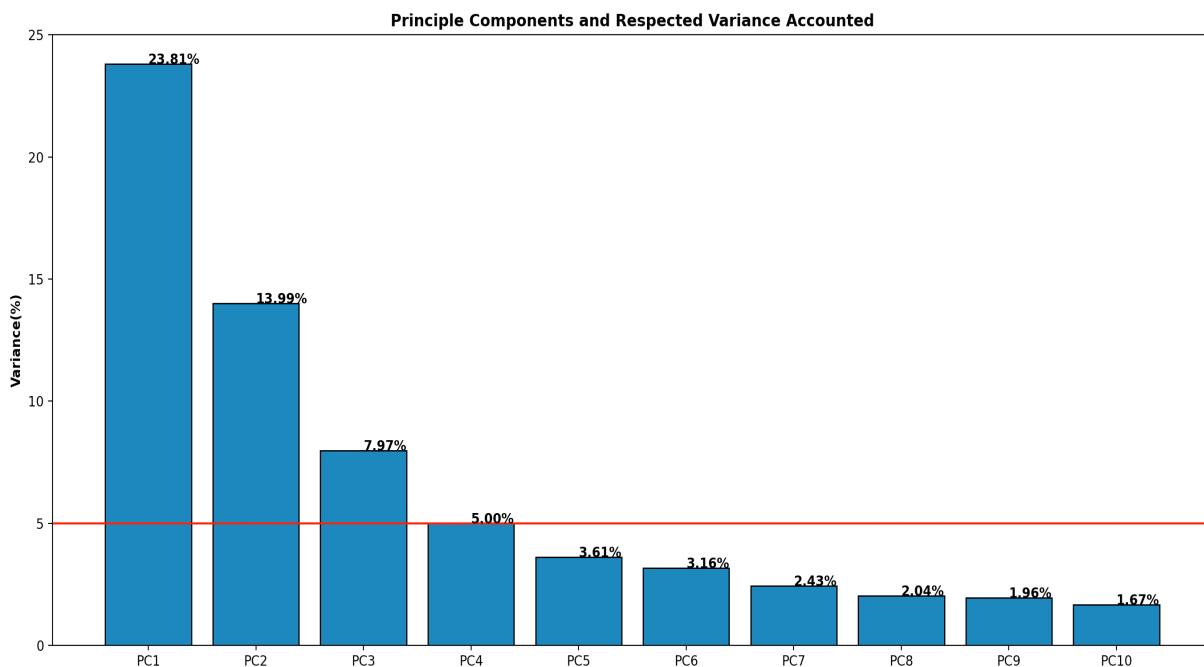


d)

400th eigenvalue is 0 means that the corresponding dimension (component) has a variance of 0 and, therefore does not exist. Essentially, n points in the n-dimensional space will be situated on a co-dimension of one affine subspace. So, the variance in the direction orthogonal to this subspace will be zero and we know that the variance can be expressed by an eigenvalue of the covariance matrix of these n points. Or in more detail, since each eigenvector corresponds to an eigenvalue, whose magnitude dictates the amount of data's variability that is explained by its eigenvector, thus the corresponding dimension (component) for that eigenvalue does not exist and this is why it is zero.

e)

For this question, I have decided to plot the graph showing the percentage of variance that is accounted by each principal component. Since it was messy to fit all principal components in one graph and components get smaller and smaller anyway, I have decided to plot the first 10 principal components.



We can see from the graph that the first 4 components account for most of the variance. The eigenvalues get smaller and smaller, thus accounting for less and less data. Therefore, the eigenvalues approaching to 0 can usually be dropped since the reconstruction error for those will be negligible and hence, very little amount of data will be lost. If we are willing to take into account all the nonzero components, then we would want to keep 399 principal components since the very last eigenvalue is zero.

However, if we want our principal components to account for 80% of the variance, we would then choose the first 28 components. (I have summed components to see what percentage of variance they account for altogether). So, for:

80% of variance -> need the first **28** principal components.

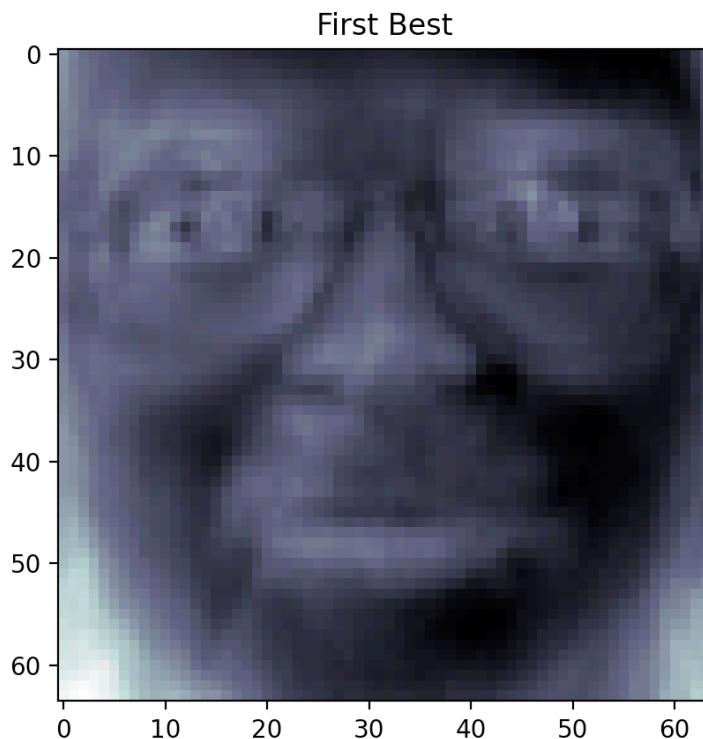
85% of variance -> need the first **40** principal components.

90% of variance -> need the first **66** principal components.

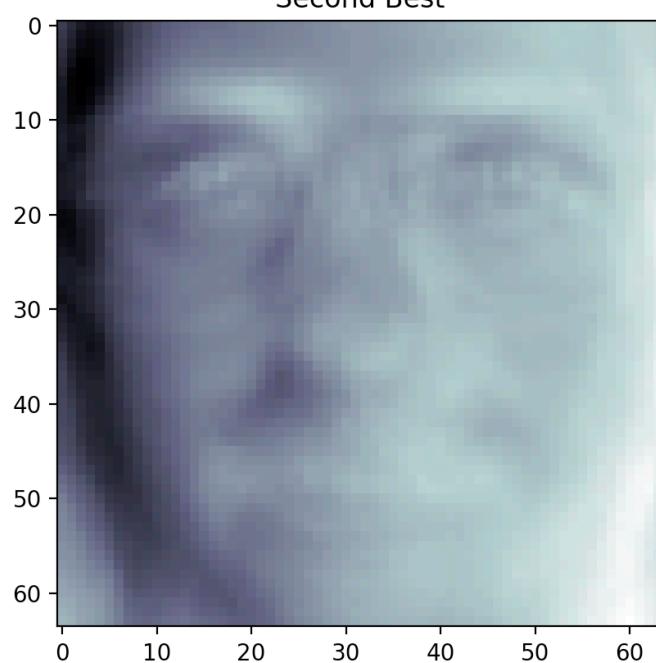
95% of variance -> need the first **120** principal components.

f)

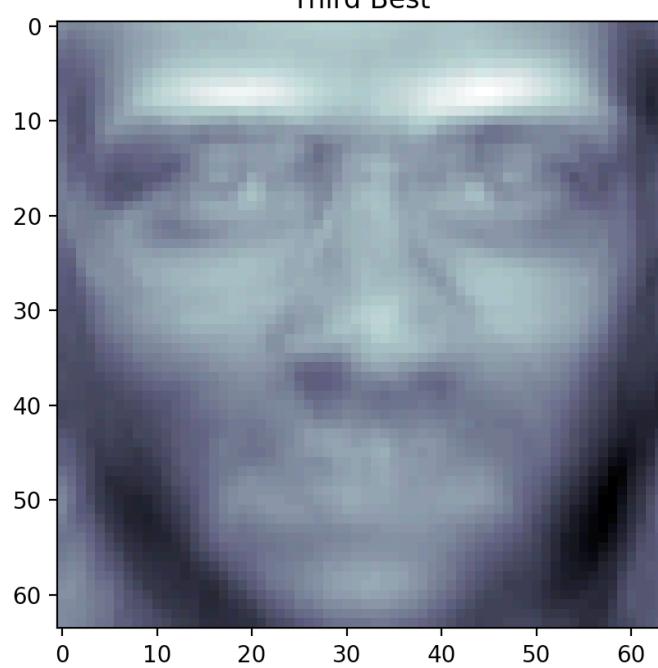
Since the eigenvectors are the eigenfaces in our situation, here are the top-5 eigenfaces corresponding to the top-5 largest eigenvalues:



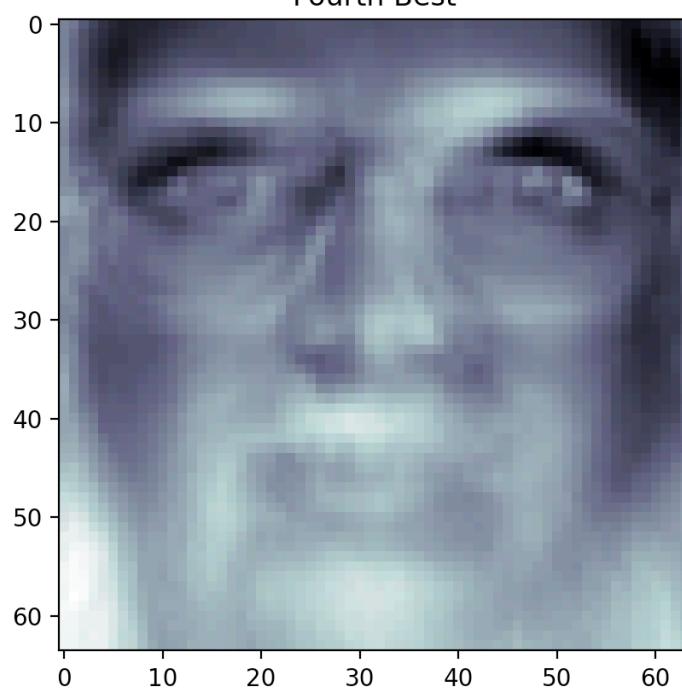
Second Best



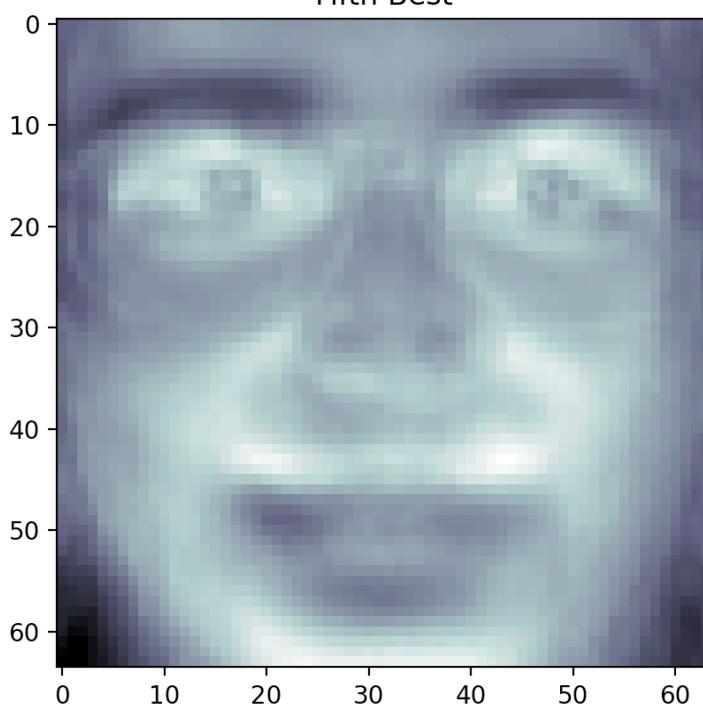
Third Best



Fourth Best



Fifth Best



g)

Here are the 4 reconstructed 100th images using 10, 100, 200 and 399 principal components:

