Survivor Face Recognition

Abstract: For a fun personal project, we decided to analyze a dataset of every contestant from the TV show Survivor and the five Computer Science professors at Florida Southern College. The goal was to answer some interesting questions, like which professor looks least like a contestant, who could most likely host the show, which season they might belong to, and who would win Survivor. To figure this out, we used algorithms like Nearest Neighbors and K-Means Clustering, along with Principal Component Analysis (PCA), to map the images into "face space" for comparison.

Loading The Data:

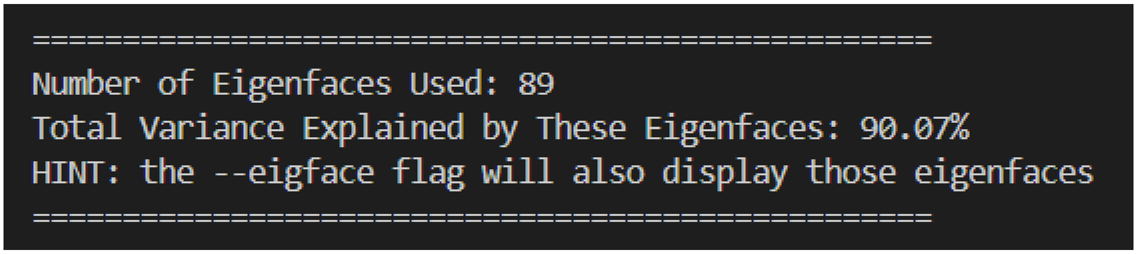
For the Ranking CSV File:
The rankings file contains the names of the Survivor contestants in the first column with the format: "Sxx_FirstName_LastName", where Sxx represents the season number. The second column contains the placement they achieved in the competition.

To process this data, we split the first column by the underscore ('_') to separate the season, first name, and last name. We then stored the full name (first and last name separatly). After that, we created a dictionary where each name is mapped to the placement they reached in the competition. This dictionary will be used later to help predict which professor is most likely to win Survivor by comparing them to contestants in the same cluster.

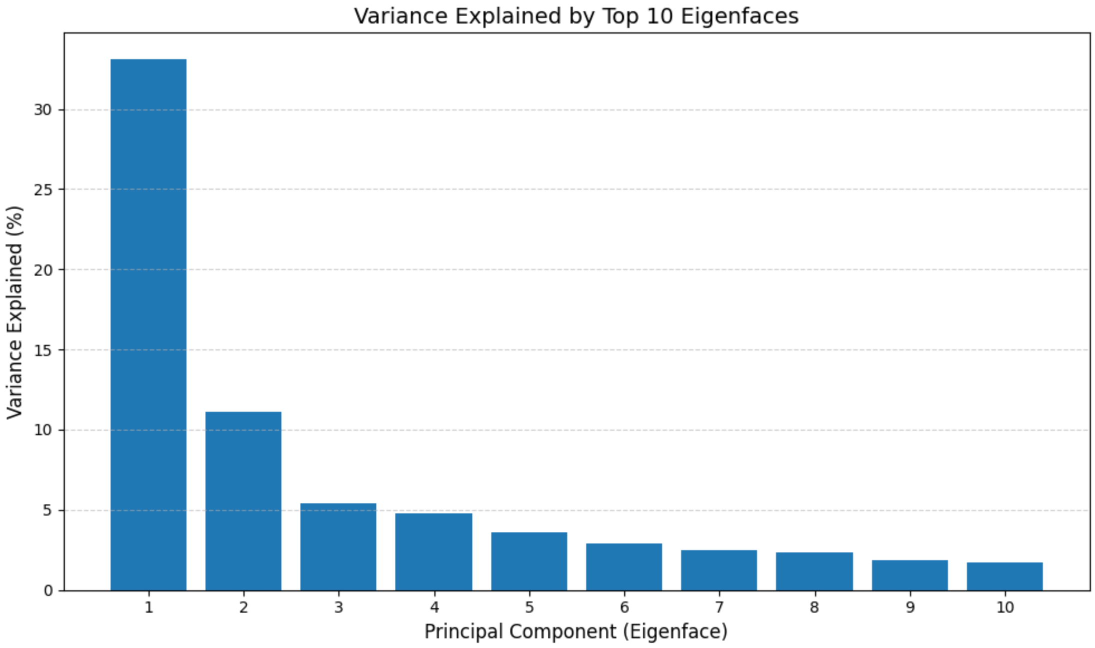
Building the PCA Model:

For this task, we set a threshold to retain 90% of the variance when reducing the dimensionality of the face images using Principal Component Analysis (PCA). We first fit the PCA model using the Survivor dataset and then transformed both the Survivor faces and the professor faces into the reduced feature space.

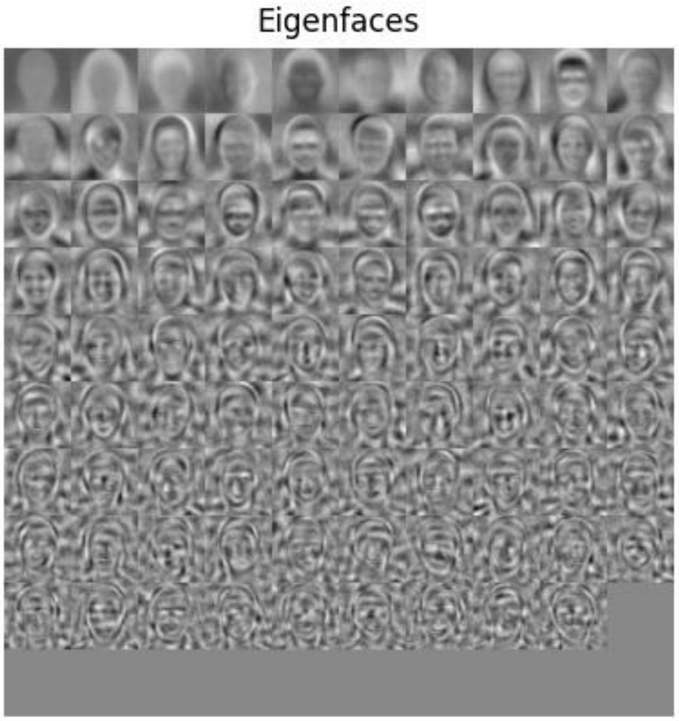
The PCA model explained 90.07% of the variance using the top 89 eigenfaces. This demonstrates the model's ability to capture a significant portion of the data with a relatively small number of eigenfaces, especially compared to the full dataset of images. We also visualized the variance explained by the top 10 eigenfaces.



```
==================================================
Number of Eigenfaces Used: 89
Total Variance Explained by These Eigenfaces: 90.07%
HINT: the --eigface flag will also display those eigenfaces
==================================================
```



Additionally, we reconstructed the images from the PCA-transformed data to visualize the results. These reconstructions provide insight into how well the PCA model captures the essential facial features while reducing the number of dimensions. The code includes several flags that allow the user to visualize interesting images generated from these reconstructions, such as the reconstructed professor and Survivor faces, as well as the Eigenfaces used for dimensionality reduction.



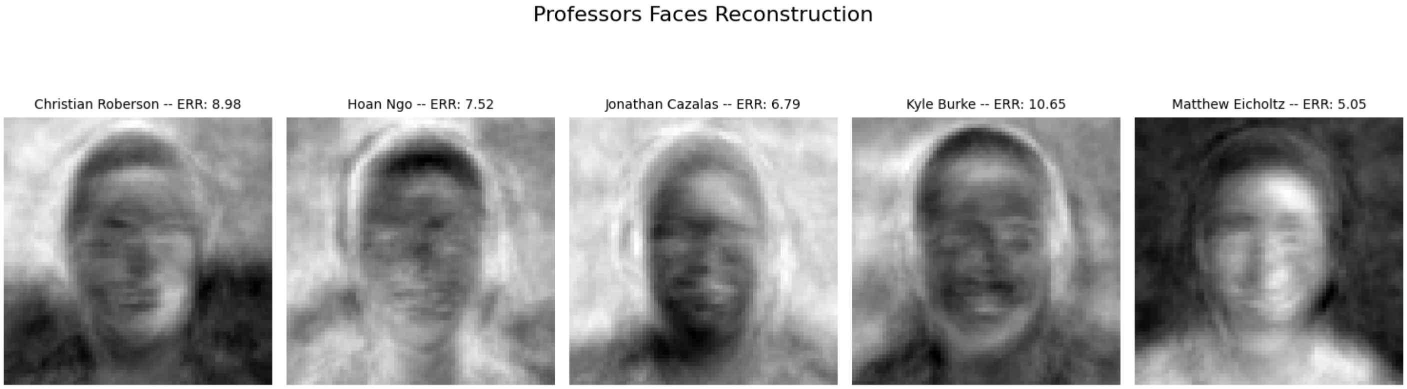
Professor that looks least like a face:

For this task, we used our PCA model to project each professor's image into the reduced "face-space." After transforming the professor faces, we reconstructed each face using the principal components learned from the Survivor dataset.

We then calculated the error for each professor by computing the difference between the original face and its reconstructed version. The error represents how much information is lost during the reconstruction process. Specifically, we measured the Euclidean distance between the original and reconstructed face for each professor.

The professor with the largest error score was considered the one who looks least like a face, as the PCA model, trained on human faces, struggled to represent their facial features accurately.

The model indicated that Dr. Burke’s face was the least face-like, with an error of 10.65. The results of the transformed faces using the PCA model are displayed below. While none of the faces were perfectly reconstructed, it is evident that the model captures around 90% of each image's variance, providing a reasonably accurate reconstruction in our view.


Professors Faces Reconstruction
Christian Roberson -- ERR: 8.98
Hoan Ngo -- ERR: 7.52
Jonathan Cazalas -- ERR: 6.79
Kyle Burke -- ERR: 10.65
Matthew Eicholtz -- ERR: 5.05

Nearest Neighbor for “who will be the next host?”:

For this task, the goal was to determine which professor was most likely to host Survivor. We first load the image of Jeff Probost, the host. We then project both his image and the professors into a higher dimensional space using PCA, which is done earlier in the code. Our goal is to create a cluster with all the professors and see which neighbor (professor) in the cluster Jeff is closest to. We simply create a Nearest Neighbors algorithm with one cluster and use the KNeighbors method on Jeff to see which professor he is closer to.

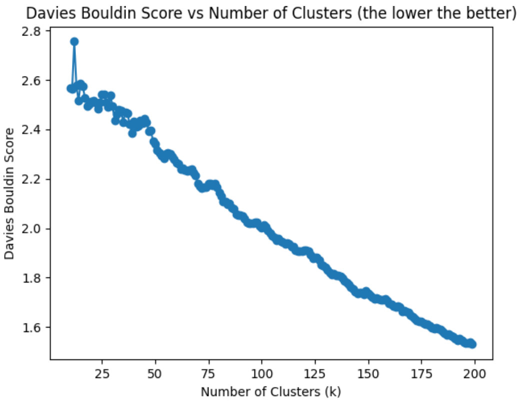
Our model predicts that Dr. Eicholtz is the most likely candidate to be the next host, as his face had the smallest distance to Jeff Probst's face, with a value of 15.78.

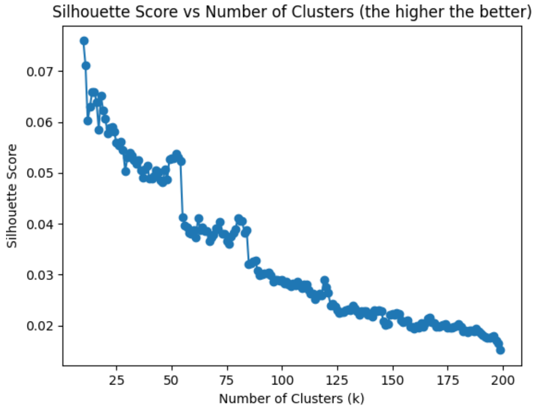
Building the K Means Cluster model & Predicting the season for each Professor Face:

For this project, we used the sklearn.cluster library to build a K-Means clustering model. The main challenge we faced was determining the optimal number of clusters. Initially, we started with 46 clusters to match the number of Survivor seasons, but we needed a way to evaluate how well this model was performing.

After training the K-Means model and assigning clusters, we created a dictionary where each cluster index was associated with a list of seasons corresponding to the Survivor faces in that cluster. To predict which season each professor would likely be on, we first predicted the cluster for the professor's face and then retrieved the list of seasons for that cluster. The season that appeared most frequently in the cluster was predicted as the season the professor would be most likely associated with.

To evaluate the model, we used built-in metrics such as the silhouette score and the Davies-Bouldin index. However, these metrics provided conflicting insights as we increased the number of clusters. While one metric improved with more clusters, the other worsened, which made it difficult to determine the ideal number of clusters based solely on these scores.



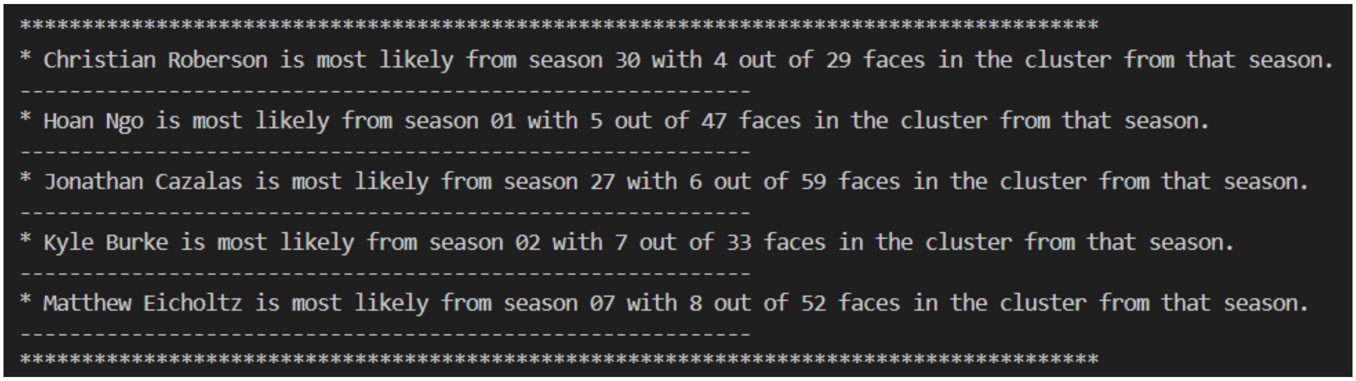


Additionally, we observed that with a larger number of clusters, many clusters contained only a small number of faces from the same season, such as two faces out of forty in a cluster. This indicated poor clustering performance, as it suggested that the model was not grouping faces from the same season effectively.

As a temporary solution, we devised a custom metric to evaluate the model's performance. We calculated the percentage of faces from the same season in each cluster and used this as a score to measure the clustering quality. This approach revealed that reducing the number of clusters—fewer than the number of seasons—improved the percentage score. After experimenting with different values, we found that using 20 clusters produced a better overall percentage score, suggesting that fewer clusters helped the model group faces more meaningfully.

By adjusting the number of clusters and focusing on the percentage of faces from the same season within each cluster, we were able to improve the predictions for which season each professor would likely be on. This approach provided a balance between clustering quality and meaningful predictions.

The model made reasonable predictions for each professor's likely season, supported by a solid number of similar faces from the same season within the predicted cluster. Our custom statistic also showed an average confidence in season prediction of 14.24% The detailed results are displayed below.



```
***************************************************************************************
* Christian Roberson is most likely from season 30 with 4 out of 29 faces in the cluster from that season.
-----------------------------------------------------------
* Hoan Ngo is most likely from season 01 with 5 out of 47 faces in the cluster from that season.
-----------------------------------------------------------
* Jonathan Cazalas is most likely from season 27 with 6 out of 59 faces in the cluster from that season.
-----------------------------------------------------------
* Kyle Burke is most likely from season 02 with 7 out of 33 faces in the cluster from that season.
-----------------------------------------------------------
* Matthew Eicholtz is most likely from season 07 with 8 out of 52 faces in the cluster from that season.
-----------------------------------------------------------
***************************************************************************************
```

Which professor is most likely to win Survivor:

For this task, we adopted a strategy similar to the one used for predicting which season each professor would likely be on. First, we utilized the dictionary created during the data loading phase, which mapped each Survivor's name to the rank or place they achieved in the competition. We combined this with the existing K-Means clustering model.

We then built another dictionary, where each cluster index from the K-Means model corresponded to a list of ranks or placements for the Survivor faces within that cluster. Essentially, for each cluster, we kept track of how well the Survivors in that group performed in the competition.

Our model predicted that Dr. Burke is most likely to win survivor, his cluster’s average being the smallest. His average place in the competition would be the 9th place.

The predicted place for each Professor is displayed below:

Place 11 – Christian Roberson
Place 10 – Hoan Ngo
Place 11 – Jonathan Cazalas
Place 9 – Kyle Burke
Place 9.5 – Matthew Eicholtz

Closing:

In this analysis, we successfully applied Principal Component Analysis (PCA), Nearest Neighbors, and K-Means clustering to tackle various tasks related to facial recognition and clustering. By leveraging these techniques, we identified which professor’s face least resembles a human face, who would be the most likely to host Survivor, and even predicted which season each professor would belong to. Our custom evaluation metrics for clustering helped refine predictions and improve the model's performance, offering meaningful insights. While further fine-tuning could enhance accuracy, our approach demonstrated the power of unsupervised learning in this unique context.

Next, we predicted the cluster for each professor using the K-Means model. For each professor, we calculated the average rank of the Survivors in the same cluster. The rationale is simple: the lower the average rank of the Survivors in a cluster, the better their performance in the competition.

The professor with the lowest average rank was considered closest to first place, meaning they would be the most likely to win Survivor. This approach allowed us to quantitatively evaluate which professor would perform best in the competition, based on the performance of Survivors with similar facial features.