Analysis Report

Data Loading and Splitting:

- The code begins by loading the Olivetti Faces dataset using fetch_olivetti_faces from Scikit-Learn.
- It splits the data into training (60%), validation (20%), and test (20%) sets using train_test_split. Stratified sampling is used to ensure a balanced distribution of target labels in each split.
- **Cross-Validation:**
- The code performs k-fold cross-validation (k=5) on the training data using a Support Vector Classifier (SVC) with a linear kernel. Cross-validation scores are stored in the score's variable.

Determining Optimal Number of Clusters:

- The code calculates the silhouette score for different numbers of clusters (ranging from 2 to 10) using K-Means clustering.
- The silhouette score helps in determining the optimal number of clusters. It measures how similar an object is to its own cluster compared to other clusters.
- The silhouette scores are plotted to help visualize and choose the optimal number of clusters.

Cluster Labels and Classification:

- The code then creates a K-Means model with the optimal number of clusters obtained from the silhouette scores.
- It initializes another SVC classifier with a linear kernel.
- A pipeline is created that first applies K-Means clustering and then applies the classifier.
- The pipeline is fitted on the training data.
- The accuracy of the classifier on the validation set is calculated and printed.

- Predictions on the validation set are made, and a classification report and confusion matrix are displayed.

DBSCAN Clustering:

- The code preprocesses the images by standardizing the feature vectors.
- It calculates the pairwise cosine distance matrix and converts it into a similarity matrix.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering is performed using the cosine similarity matrix. DBSCAN is a density-based clustering algorithm.
- The cluster labels are printed, as well as the cosine similarity and cosine distance matrices.
- Optional visualizations of the cosine similarity matrix and cluster labels in a scatter plot are provided.

Outputs:

- The cluster labels in DBSCAN are mostly assigned to a single cluster (label 0). This suggests that DBSCAN may not be effectively clustering the data.
- The classification results using K-Means and the SVC classifier show low accuracy (approximately 0.28), indicating that the current approach may not be suitable for this dataset.

Suggestions for Improvement:

- You may need to explore different clustering and classification algorithms or consider alternative feature extraction methods for better results.
- Hyperparameter tuning for the classifiers and clustering algorithms could improve performance.
- It's important to preprocess and extract meaningful features from the images for better clustering and classification results.