Purpose of PCA PCA is a dimensionality reduction technique that helps to:

- Reduce the number of features while preserving most of the important information
- Identify the most significant components that explain the variance in the data
- Simplify the data representation for machine learning models

Code Breakdown

```
from sklearn.decomposition import PCA

X_full = np.row_stack((X_positive_bright, X_negative_bright))
```

- This combines two datasets (positive and negative samples) into a single array

```
pca_full_variance = PCA(n_components = 0.95)

pca_full_variance.fit(X_full)
```

- Creates a PCA object that will retain enough components to explain 95% of the data's variance
- `fit()` method calculates the principal components

```
pca_full_variance.components_.shape
```

- Returns (96, 10000), meaning 96 principal components were needed to explain 95% of the variance
- Each component is a vector of 10,000 dimensions (likely representing pixels in an image)

3. **Visualization** The code creates a plot of the first 24 principal components, visualizing them as grayscale images. The interpretation suggests:

- First component shows characteristics of uninfected samples (lighter colors)
- Second component shows characteristics of parasitized samples (darker colors)

4. **Transformation**

```
X_reduced = pca_full_variance.transform(X_full)

X_reduced.shape
```

- `transform()` projects the original data onto the principal components
- Reduces the original high-dimensional data (likely 10,000 features) to 96 features
- Output shape is (27558, 96) - maintaining the number of samples but reducing feature dimensions

Key Benefits:

- Reduced computational complexity
- Removed redundant or less important features
- Prepared data for machine learning models like Support Vector Machines (SVM)

In essence, PCA helps you compress the data while retaining its most important characteristics, making it easier and more efficient to train machine learning models.