# CS 191 Principal Component Analysis (LFW Face Recognition)

Submitted by
Femo Bayani and Mikaela Ramos

## INTRODUCTION

### Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a well-known and widely-used technique for dimensionality reduction, which is the considerable reduction of the number of variables while still retaining much of the information of the original data set. [1]

## IMPLEMENTATION

### Dataset

The dataset for this exercise is the Labeled Faces in the Wild (LFW) dataset which is obtained using scikit-learn's `sklearn.datasets.fetch_lfw_pairs()` method. This returns a dataset of 13233 total images from 5749 different people. The X values correspond to a pair of two images, with the y value referring to either "same person" or "different person".

### Principal Component Analysis

The implementation used for performing PCA is `sklearn.decomposition.PCA`, with a specified seed for the random state for consistent results. Whitening is also performed by PCA.

```
pca = PCA(n_components=n_components, whiten=True,
          random_state=rnd_state).fit(X_train)

# Transform the dataset using PCA
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
```

Additionally, the cumulative variance is obtained as such:

```
pca = PCA(whiten=True, random_state=rnd_state).fit(X_train)
cumvar = np.cumsum(pca.explained_variance_ratio_)
```

### SVM

Then, after dimensionality reduction by the PCA, the transformed data is used to train an SVM classifier, particularly `sklearn.svm.SVC`. The gamma is set to auto, but all the other parameters are set to default, such as the kernel being rbf.

```
# Train SVM
clf = SVC(gamma='auto', random_state=rnd_state)
clf = clf.fit(X_train_pca, y_train)
```

After training, the classifiers predicts the test set.

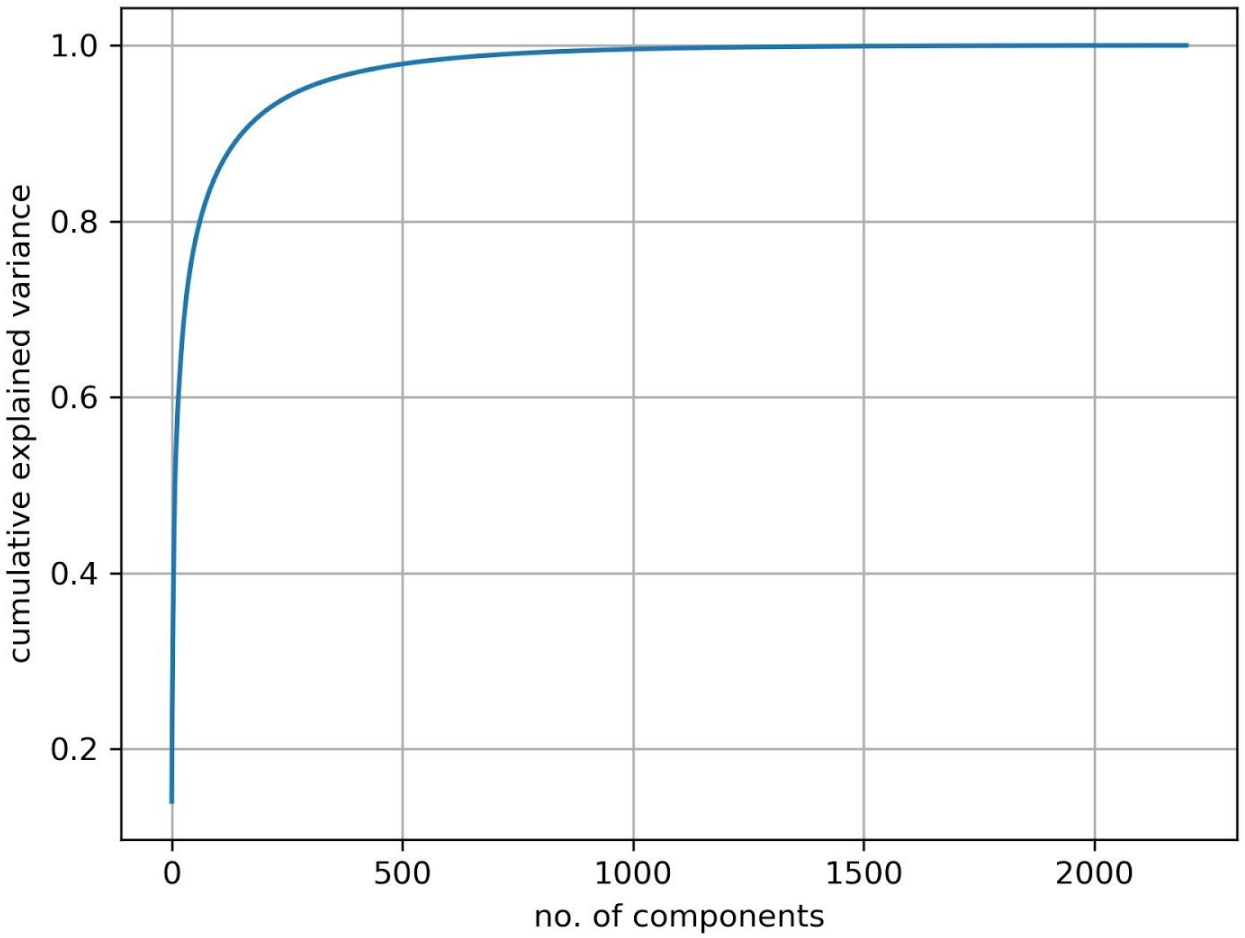
```
y_pred = clf.predict(X_test_pca)
```

Then, the accuracy is obtained using `sklearn.metrics.accuracy_score`, which is a utility method for obtaining accuracy.

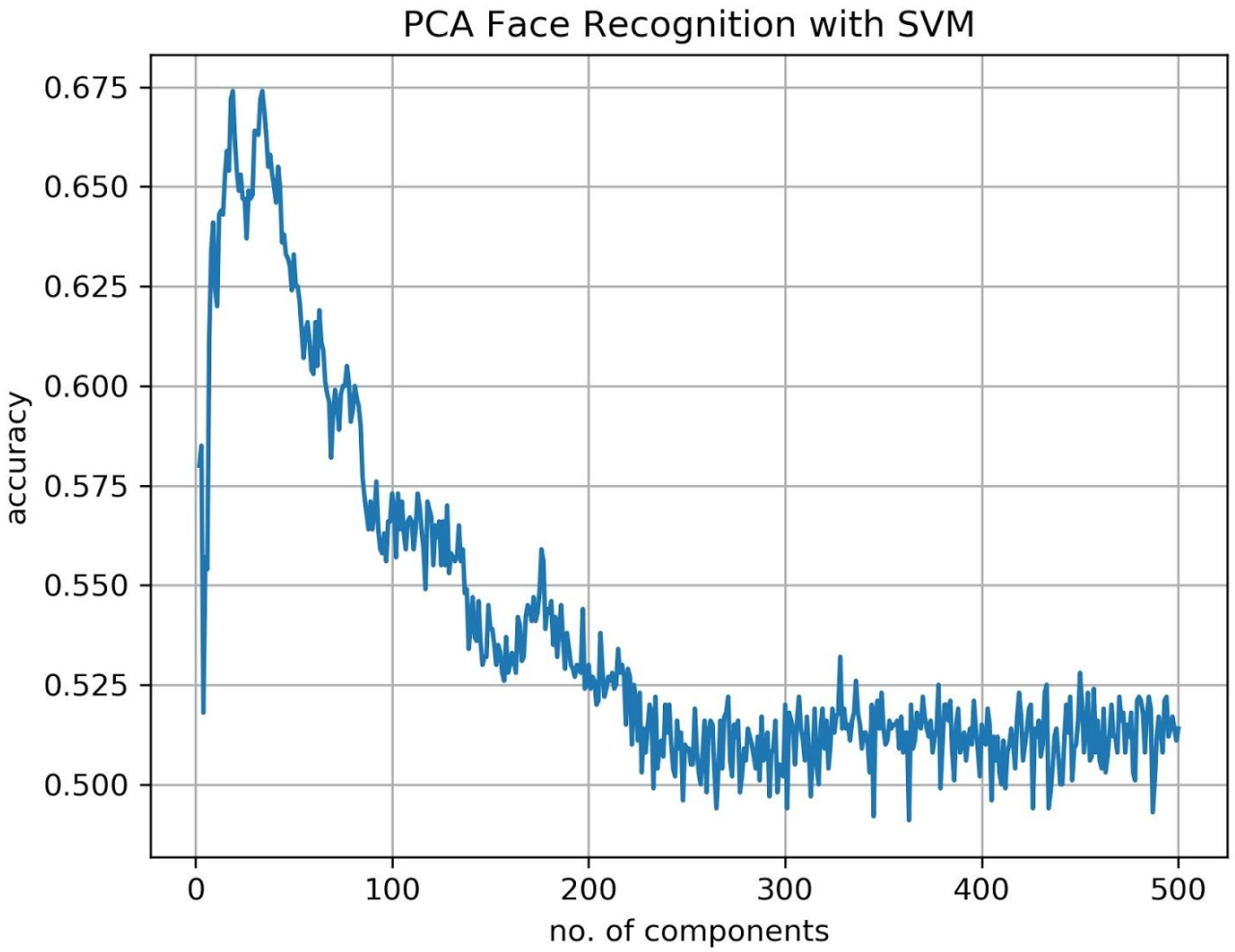
```
accuracy = accuracy_score(y_test, y_pred)
```

## RESULTS

Regarding the plot of the cumulative variance:



Then, the results for the comparison of the number of components vs the accuracy of the SVM.




We can see on the graph that it starts it relatively low then climbs up until a maximum, then it starts to have a steep downward trend, until it mostly stagnates (while oscillating). It reached the maximum accuracy at 19 components and 34 components, both with 67.4% accuracy.

## CONCLUSION

The accuracy of 67.4% isn't considerably big, it's actually pretty low as an accuracy, but take note that no hyperparameter optimization was performed for the parameters of the SVM model, which may have affected the resulting accuracies. However, the trend of the effect of the number of components vs the accuracy is shown and shows that there's a an accuracy can rise up to a maximum point, at which it'll drop from there.

## REFERENCES

[1] Jolliffe I. (2011) Principal Component Analysis. In: Lovric M. (eds) International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg