Facial recognition using machine learning methods

Introduction

Machine learning is a technology which involves the development of algorithms and statistical models that enable the machine to learn from past data. The goal of machine learning is to identify patterns and relationships in data. Machine learning is a rapidly growing field and has been used for a variety of tasks. Facial recognition is a biometric identification technology that uses the unique features of an individual's face to identify them, most facial recognition systems work by comparing facial impressions with a database of known faces. In this article we'll use some powerful models to do the facial recognition.

Method 1: K-Means Clustering

K-Means clustering algorithm is an unsupervised machine learning method which can identify clusters of objects in a set. There are many different types of clustering methods, K-means is the most easily understood method. This algorithm first divides the data into K groups, randomly selects K objects as the initial cluster centers, and then calculates the distance between each object and each seed cluster center, assigning each object to the cluster center closest to it. After iterations, the cluster centers and the objects assigned to them represent a cluster. That is, K-Means algorithm takes some columns of the input table as features and clusters the original data into several classes according to the user-specified similarity calculation.

Algorithm 1 k-means algorithm

1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.

Method 2: PCA (Principal component analysis)

Principal Component Analysis (PCA) is a mathematical technique used for reducing the dimensionality of a dataset. It transforms a set of correlated variables into a new set of uncorrelated variables, called principal components, in order to capture the most important features of the data in a smaller dimension. Principal components are linear combinations of the original variables and can have no more dimensions than the original data. The result of this transformation is a new set of observations that retain the key features of the original data, but are expressed in a lower dimension for easier analysis. These new variables are uncorrelated, which makes them ideal for presenting the characteristics of the data in a simplified form.

PCA application in facial recognition

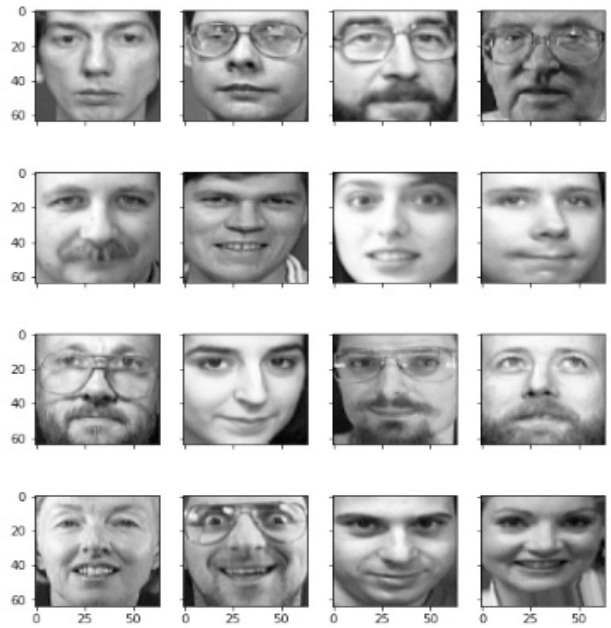
We can take a look at a simple example. The dataset used here is called" AT&T Database of Faces"

```
faces = datasets.fetch_olivetti_faces()
print("Flattened Face Data shape:", faces.data.shape)
print("Face Image Data Shape:", faces.images.shape)
print("Shape of target data:", faces.target.shape)
```

```
Flattened Face Data shape: (400, 4096)
Face Image Data Shape: (400, 64, 64)
Shape of target data: (400,)
```

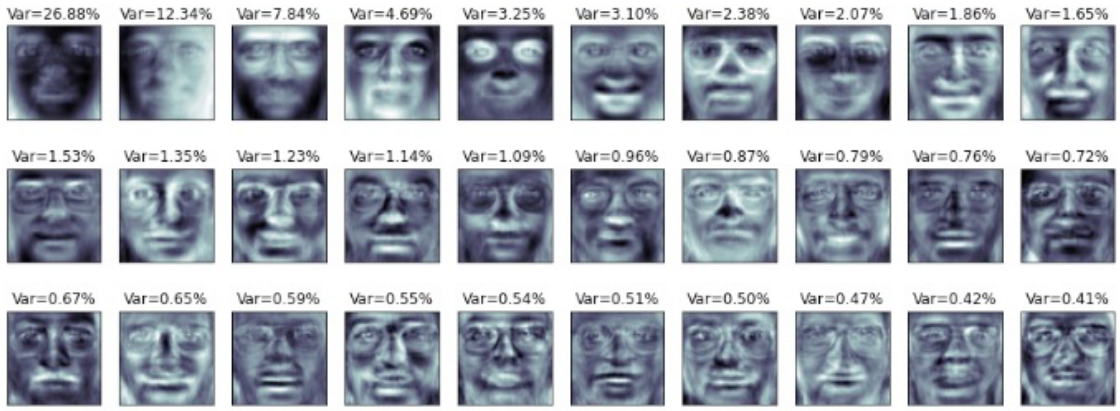
```
#Prints some example faces
faceimages = faces.images[np.random.choice(len(faces.images),size= 16, replace = False)]

fig, axes = plt.subplots(4,4,sharex=True,sharey=True,figsize=(8,10))
for i in range(16):
    axes[i%4][i//4].imshow(faceimages[i], cmap="gray")
plt.show()
```



We extract the images from the dataset and each image is 64 by 64 image with pixels. Above sixteen graphs gives how these data look like.

```
fig = plt.figure(figsize=(16, 6))
for i in range(30):
    ax = fig.add_subplot(3, 10, i + 1, xticks=[], yticks=[])
    ax.imshow(pca.components_[i].reshape(im_shape),
              cmap=plt.cm.bone)
    ax.set_title(f"Var={pca.explained_variance_ratio_[i]:.2%}")
```



In this example, we apply PCA analysis to the entire data and print out the 30 most relevant features to the original data. We can see that the image with the most relevant variance of 26.88% is an image of a blurred human face. In face recognition systems, PCA is used to convert a face image into a low-dimensional representation called a eigenface. Eigenfaces capture the most important features of the face, such as the shape of the eyes, nose, mouth, and other facial features, and discard less important information, such as texture and illumination. When a new face image is presented for recognition, the system first converts it into the eigenface space and compares it with the eigenfaces of known faces. The recognition process involves finding the closest match between the new face image and the known faces. The face image that is closest to the new image is considered to be the recognized face. PCA is an effective face recognition technique because it reduces the dimensionality of the data while preserving the important features of the face, making the recognition process more efficient and accurate.

The combination of PCA and K-means

```
from sklearn.cluster import KMeans

n_clusters = 10 #We know there are 10 subjects
km = KMeans(n_clusters  = n_clusters,random_state=0)

pipe= Pipeline([("scaler",StandardScaler()), #First standardize
                ("pca",PCA()), #Transform using pca
          ("kmeans", km )]) #Then apply k means
clusters = pipe.fit_predict(faces.data)

for labelID in range(n_clusters):
    # find all indexes into the `data` array that belong to the
    # current label ID, then randomly sample a maximum of 25 indexes
    # from the set
    idxs = np.where(clusters == labelID)[0]
    idxs = np.random.choice(idxs, size=min(25, len(idxs)),
        replace=False)

    # Extract the sampled indexes
    id_face = faces.images[idxs]

    #Plots sampled faces
    fig = plt.figure(figsize=(10,5))
    for i in range(min(25,len(idxs))):
        ax = fig.add_subplot(5, 5, i + 1, xticks=[], yticks=[])
        ax.imshow(id_face[i],
                  cmap=plt.cm.bone)
    fig.suptitle(f"Id={labelID}")
```

Now we combine PCA and K-Means.In this code, we normalize the data, then put it through the PCA transformation, and finally use the K-Means model. Finally we can get ten clusters. By eye, we can find that each image in each cluster has many similarities with other images in the same group.

Conclusion

In summary, Principal Component Analysis (PCA) and K-Means are two techniques commonly used in facial recognition systems. PCA can reduce the dimensionality of a face image and retain important features, and can convert a high-dimensional face image into a low-dimensional representation called eigenface, which captures the most important features of a face. k-Means is used to cluster similar faces and perform the actual recognition process. k-Means is used to group similar faces into clusters, with each cluster representing a separate face class. When a new face image is presented for recognition, the system compares it with the eigenfaces of known faces and assigns it to the category that matches the closest.

In summary, the combination of PCA and K-Means provides an effective and efficient solution for face recognition. the dimensionality reduction provided by PCA helps to reduce the complexity of the recognition process, while the clustering provided by K-Means helps to group similar faces and perform the actual recognition.

Reference

1. Kasikrit Damkliang. 2020. AT&T database of faces. https://www.kaggle.com/datasets/kasikrit/att-database-of-faces
2. Education Ecosystem.2018. Understanding K-means Clustering in Machine Learning. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
3. UCLA M148 by Professor Dolecek