IDC-409 Project-2
Signature Fraud Busting

Group members - Rimjhim Goel (MS18133), Chhavi Chahar (MS18136)

1 Contribution

Rimjhim Goel (MS18133) - 50 percent and Chhavi Chahar (MS18136) - 50 percent

2 Problem

You have a collection of signatures. Given a new signature, you are asked to find out if it is a real signature or a fake one. Write a possible solution using K-means or any other clustering method.

3 Source of dataset

The data has been taken from Kaggle.

The dataset available on Kaggle for signature verification includes a collection of authentic and forged signatures of a number of people which can be used for training and testing the algorithm.

4 Data pre-processing

We have used tensorflow and keras for pre-processing the images.

Inception v3 assigns weigths pre-trained in ImageNet data for extracting the image features. The ImageNet data set is a collection of images belonging different categories and labels on which the module has been trained.

After extracting the features of each image, we extracted arrays from each image. The arrays were then pre-processed. After training, we used predict() to predict the labels and the data was flattened to 1 dimension.

We created a pandas data frame to show the name of each image with its corresponding cluster.

5 Approach to the problem

The algorithm used is k-means.

Since in the given problem, we had to use clustering methods and not the supervised learning methods, we took the the collection of signatures of a single person. The data set contained both - the authentic signatures and the forged ones. We were aware which signatures were authentic as given in the problem.

On the basis of this information, we try to apply k means on the data with k=2.
Since we already knew which data points are the authentic signatures, by looking at the clusters we could identify the forged signatures.

For testing the algorithm, we first supplied only the authentic data and observed that the algorithm clustered all the points in a single cluster indicating there was no significant difference between them.

After the testing, we ran the algorithm on a collection of authentic and forged data sets. The algorithm clustered the two separately. Since we were already aware which were the authentic ones, the problem of forged signature busting was solved.

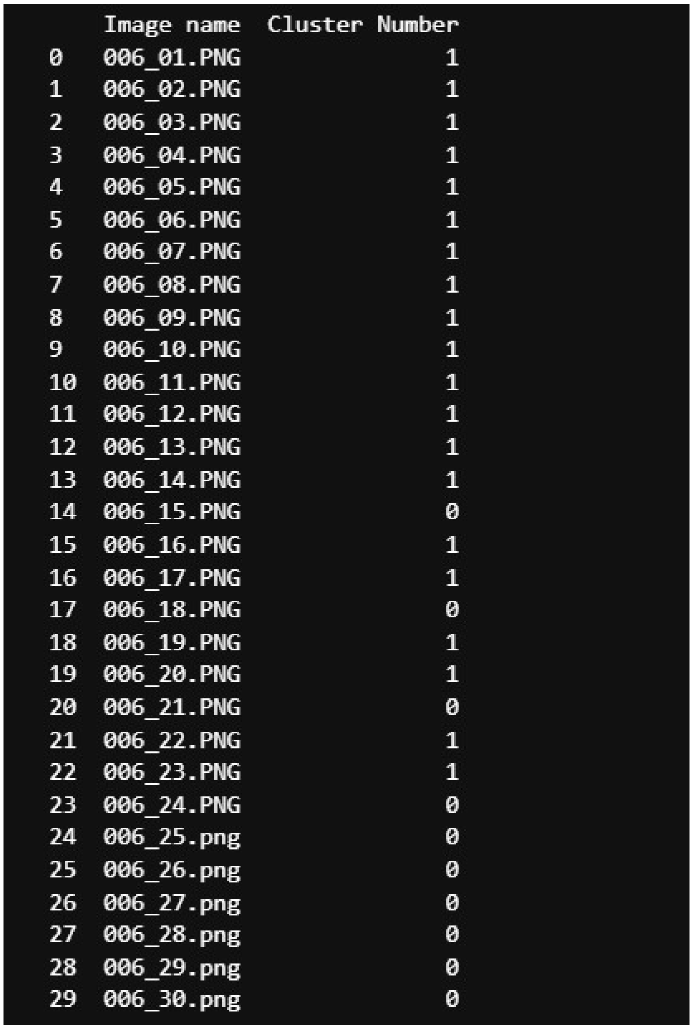
6 Results

6.1 The code for the problem is as follows -

```
#Importing the required libraries

from sklearn.cluster import KMeans
import os
import pandas as pd
import numpy as np
from tqdm import tqdm

from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.applications.inception_v3 import preprocess_input

#Function for  pre-processing and extracting the image features

def image_features(directory):
    train = InceptionV3(weights='imagenet', include_top=False)
    #for pre-processing, pre defined weights used for identifying the object from the image
    features = []  # list of features extracted
    image_name = []  # the image names
    for i in tqdm(directory):
        file_name=''+i   #  file name format in which images are present in the directory
        img=image.load_img(file_name,target_size=(224,224))
        x = img_to_array(img) #converting the image to array
        x=np.expand_dims(x,axis=0) # expanding the array for adding cluster information
        x=preprocess_input(x) # pre-processing the arrays
        feature=train.predict(x) # testing
        feature=feature.flatten()  # dimension reduction
        features.append(feature) # adding the features to the list
        image_name.append(i) #adding the image name to the list
    return features,image_name
```

```

#Getting the images from the directory, extracting the features and making the dataframe

file_path = os.getcwd() # the working directory
#print(path)
image_path=os.listdir('006')  #the files are named as '006..'
#print(img_path)
#extracting the features using the function defined in previous cell
image_features,image_name=image_feature(image_path)
# creating the pandas dataframe for the images
image_cluster = pd.DataFrame(image_name,columns=['Image name'])
#print(image_cluster)

#Applying k means

k = 2 #number of clusters to be formed
''' initial state for each iteration is chosen randomly
    max. number of iterations allowed for a single run = 10000000
    tolerance in clustering  = 1e-15
    Number of times the algorithm is run with different centroids = 500

'''
clusters = KMeans(k, random_state =None,max_iter=10000000,tol=1e-15,n_init=500)
clusters.fit(image_features) #computing the clusters from the extracted features
X = clusters.fit_transform(image_features)  # computing the distance from centroids
print(X)

#Visualising the results

image_cluster["Cluster Number"] = clusters.labels_ #adding labels to each cluster
print(image_cluster)
```

6.2 The following results were obtained:



```
     Image name  Cluster Number
0   006_01.PNG               1
1   006_02.PNG               1
2   006_03.PNG               1
3   006_04.PNG               1
4   006_05.PNG               1
5   006_06.PNG               1
6   006_07.PNG               1
7   006_08.PNG               1
8   006_09.PNG               1
9   006_10.PNG               1
10  006_11.PNG               1
11  006_12.PNG               1
12  006_13.PNG               1
13  006_14.PNG               1
14  006_15.PNG               0
15  006_16.PNG               1
16  006_17.PNG               1
17  006_18.PNG               0
18  006_19.PNG               1
19  006_20.PNG               1
20  006_21.PNG               0
21  006_22.PNG               1
22  006_23.PNG               1
23  006_24.PNG               0
24  006_25.png               0
25  006_26.png               0
26  006_27.png               0
27  006_28.png               0
28  006_29.png               0
29  006_30.png               0
```

Figure 1: Clusters

7 Interpretation

Image number 01 to 23 were authentic signatures and image number 24 to 30 were forged.

Most of the authentic images have been in one cluster and the forged ones in the other cluster.

However, three of the authentic ones - 15, 18 and 21 have been declared fake by the algorithm. Signatures have a possibility of differing from each other even when they are done by the same person. There may be subtle differences and the algorithm might be catching those.

No fake signature has been declared authentic by the algorithm.

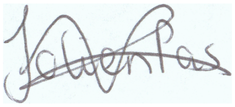


Figure 2: Authentic signature

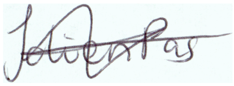


Figure 3: Fake signature identified as authentic

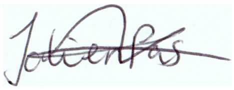


Figure 4: Fake signature identified as authentic

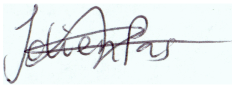


Figure 5: Fake signature identified as authentic

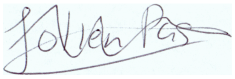


Figure 6: Fake signature identified as fake

8 List of files attached

1. Code
2. The images used for clustering