BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Machine Learning Assignment - 2

Ethnicity/Nationality Identification

GROUP MEMBERS:

RIGVITA SHARMA	2016A7PS0067P
PRACHI AGARWAL	2016A2PS0769P
SHRUTI SHARMA	2016ABPS0833P

Index

Introduction

Problem Statement - Trying to establish the Ethnicity of a person by just looking at his facial image.

The significance of this program -

- Identification of ethnicity could help in increasing the focus on identity-related features thus decreasing the computational power to increase the computational speed. This might prove to be helpful in forensics for the identification of culprits.
- It might help suppliers to develop a deeper understanding of individuals and would help them to bridge the gap between them. This plays a critical role in developing a insight and more precise information about an individual's demands. It will help to cater to the products according to the demands of under-represented sections.
- To study and identify the native culture of a place, which might help in performing some social experiments.

We have tried to solve the given problem statement by using the following methods :

KNN
KNN can be used for both classification and regression predictive problems
KNN algorithm fairs across all parameters of considerations. It is commonly used for its ease of interpretation and low calculation time.
Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.
The model structure doesn't make any assumptions on underlying data distribution.
Problems:
The algorithm stores all the data therefore computationally hence computationally taxing and requires a lot of memory.
There is a possibility of slow prediction.

CNN - Convulated Neural Network
CNN or ConvNet is a class of feed-forward artificial neural network mostly used to image analysis. The connectivity between neurons resembles with the organization of animal visual cortex.

Their independence from prior knowledge and human effort in feature design is a major advantage.

SVM - Support Vector Machine
It is a classical supervised learning method used to separate data points in the high-dimensional space.

Formulation of the Problem as a machine learning model

Improving at some task with experience - Learning.
The general definition of a machine learning problem is -
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".
Here,
Task-T = To classify a set of facial images into four categories - Asian, Black, Latino, White.
Experience-E = The set of labelled data, which maps images to their ethnicities - training data.
Performance-P = Classification accuracy, number correctly predicted ethnicities - test data.

Pre-processing

Face Attribute Dataset Collection - Chicago Face Database.
We used Chicago Face Database - Version 2.0.3 - July 2016, for training our machine learning models. It includes - 597 photographs, of men and women with a neutral expression. The aim of the model is to classify the ethnicity of people into Asian, Black, Latino, White.

Image Preprocessing -

1. Cropped the face using Haar Cascade Face Detection Algorithm packaged with OpenCV to eliminate the background noise.
2. Normalize the faces - make their dimensions equal. Here, we have used resized function packaged with OpenCV - to eliminate the noise from image background.

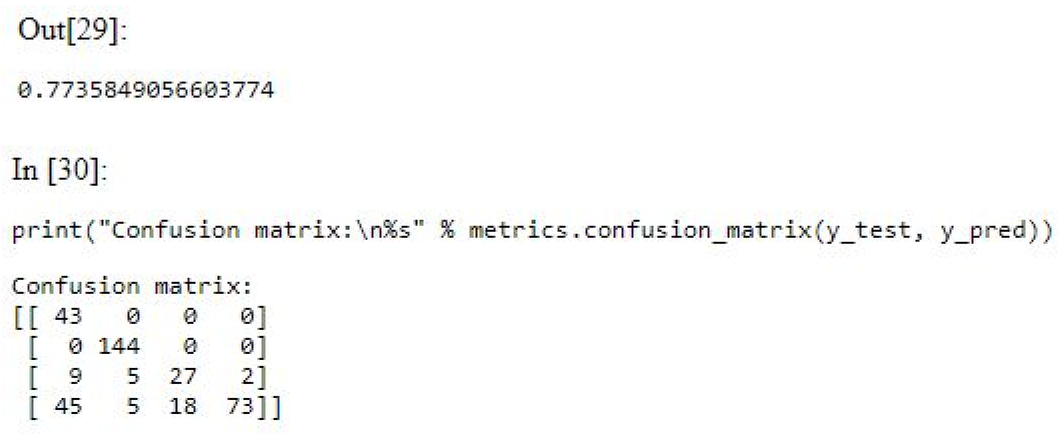
3. The reframing of images into 64*64 used for images CNN, and 128*128 used for KNN and SVM, to improve the computational speed.
4. Converted images to grayscale, to eliminate the differences of brightness and contrast in the inputs.
5. Mean subtraction - subtracting the mean across every data point.
6. Data-augmentation - flipping each image.
7. After pre-processing, we get 2,047 images.

Implementation -

KNN - K-Nearest Neighbour algorithm
KNN is used for both classification and regression predictive problems. It is more preferably used in classification problems, because of its ease of interpretation of the output, low calculation time and predictive power.

Input - 85% of the given images are used as training data, and remaining images are used for testing data;
Output - y = label of given test data;
For each image belonging to an ith test point, find the labels of these neighbours, find the most common one and this label is assigned to the labelled data.

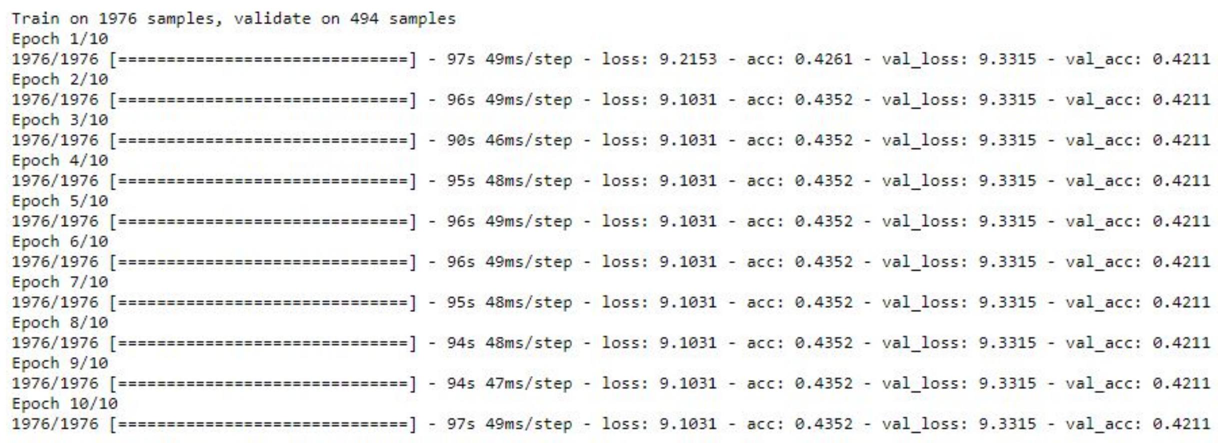
Results -
KNN
Accuracy -



```
Out[29]:

0.7735849056603774

In [30]:

print("Confusion matrix:\n%s" % metrics.confusion_matrix(y_test, y_pred))

Confusion matrix:
[[ 43   0   0   0]
 [  0 144   0   0]
 [  9   5  27   2]
 [ 45   5  18  73]]
```

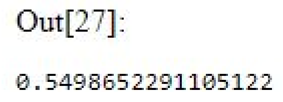
CNN
Here we tried to improve the accuracy by tuning of hyperparameters such as learning rate and regularization strength.

Results -
Training set - 1976 samples, Validation set - 494 samples

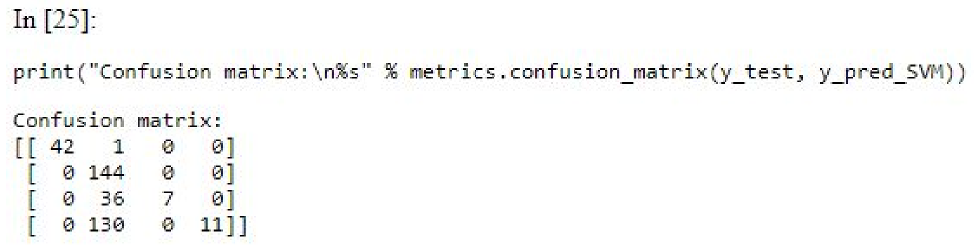


```
Train on 1976 samples, validate on 494 samples
Epoch 1/10
1976/1976 [==============================] - 97s 49ms/step - loss: 9.2153 - acc: 0.4261 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 2/10
1976/1976 [==============================] - 96s 49ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 3/10
1976/1976 [==============================] - 90s 46ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 4/10
1976/1976 [==============================] - 95s 48ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 5/10
1976/1976 [==============================] - 96s 49ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 6/10
1976/1976 [==============================] - 96s 49ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 7/10
1976/1976 [==============================] - 95s 48ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 8/10
1976/1976 [==============================] - 94s 48ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 9/10
1976/1976 [==============================] - 94s 47ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
Epoch 10/10
1976/1976 [==============================] - 97s 49ms/step - loss: 9.1031 - acc: 0.4352 - val_loss: 9.3315 - val_acc: 0.4211
```

SVM
Accuracy -



```
Out[27]:

0.5498652291105122
```



```
In [25]:

print("Confusion matrix:\n%s" % metrics.confusion_matrix(y_test, y_pred_SVM))

Confusion matrix:
[[ 42   1   0   0]
 [  0 144   0   0]
 [  0  36   7   0]
 [  0 130   0  11]]
```

Conclusion -

The accuracy of CNN is around 42% while that of KNN is 73%. The lower accuracy of the deep learning model is due to the smaller data set used for the problem solving which is due to the limited computational power. SVM is classifying the images correctly on the basis of their colour, so is able to classify the properly only half of the images most of which are based on colour. So, it has lower accuracy than all the other models.