

Practical-1

Objective:

Design a character recognition system using basic python

Description:

Given a set of 10 characters, identify the features which will distinguish each character in the character set most accurately.

Character set:

{ B , D , E , F , H , K , L , M , N , P }

Features:

1. Number of vertical lines
2. Number of horizontal lines
3. Number of diagonals
4. Number of curves

Alpha	V Lines	H Lines	Diagonal	Curve
B	1	0	0	2
D	1	0	0	1
E	1	3	0	0
F	1	2	0	0
H	2	1	0	0
K	1	0	2	0
L	1	1	0	0
M	2	0	2	0
N	2	0	1	0
P	1	0	0	1

Algorithm:

1. Read the dataset which is hardcoded in csv file using pandas.
2. Define a function which will calculate Euclidean distance for unknown/new character by using below formula .

$$\text{dist}(\text{new}, \text{compare}) = \text{square root}(\sum (\text{new}[i] - \text{compare})^2).$$

3. For new character calculate the Euclidean distance for each character present in dataset and output the one which has least distance.

Modified dataset after preprocessing:

Alpha	V Lines	H Lines	Curve	closed region
B	1	0	2	2
D	1	0	1	1
E	1	3	0	0
F	1	2	0	0
H	2	1	0	0
K	1	0	0	0
L	1	1	0	0
M	2	0	0	0
N	2	0	0	0
P	1	0	1	1

Implementation:

1 import dataset

```
import pandas as pd
features = pd.read_csv('features.csv')
```

2 calculate Euclidean distance

```
import math
def predict(new_char, features):
    min_dist = 99
    best_char = []
    count = features.shape[0]
    for i in range(count):
```

```

s = [(features.loc[i][j+1]-new_char[j])**2 for j in range(features.shape[1]-1)]
eDist = math.sqrt(sum(s))
#print(eDist)
if( eDist <= min_dist ):
    if(eDist == min_dist):
        best_char.append(i)
    else:
        best_char.clear()
        best_char.append(i)
    min_dist = eDist
return [features.loc[i][0] for i in best_char]

# 3 input new char

input_data = list()
for i in features_new.columns[1:]:
    input_data.append(int(input(str(i)+' : ')))

# 4 output

print(predict(input_data,features_new))

```

Output:

Actual Char	Before post-processing		After post-processing	
	Input(V,H,D,C)	Output	Input(V,H,D,C)	Output
F	[1,2,0,1]	F	[1,2,0,0]	F
L	[1,1,0,1]	D,L,P	[1,1,0,0]	L
M	[2,0,0,2]	B	[2,0,0,0]	M,N