Test.py

import os

import pickle

from random import random

import numpy as np

from scipy.io.wavfile import read

from featureextraction import extract\_features

#from speakerfeatures import extract\_features

import warnings

warnings.filterwarnings("ignore")

import time

import random

from stscreen import gui

from recaudio import rec

from spttex import SpeakText

"""

#path to training data

source = "development\_set/"

modelpath = "speaker\_models/"

test\_file = "development\_set\_test.txt"

file\_paths = open(test\_file,'r')

"""

#path to training data

source = "SampleData/"

accountdb=[{}]

#path where training speakers will be saved

modelpath = "Speakers\_models/"

models=[]

gmm\_files = [os.path.join(modelpath,fname) for fname in

os.listdir(modelpath) if fname.endswith('.gmm')]

"""for fname in gmm\_files:

with open(fname,'rb') as fl:

response = pickle.load(fl,encoding='bytes')

print(response)"""

#Load the Gaussian gender Models

models = [pickle.load(open(fname,'rb')) for fname in gmm\_files]

speakers = [fname.split("/")[-1].split(".gmm")[0] for fname

in gmm\_files]

for i in range(len(speakers)):

accountdb.append({"name":speakers[i],"balance":random.randint(50000,200000)})

accountdb.pop(0)

print(accountdb)

error = 0

total\_sample = 0.0

print ("Do you want to Test a Single Audio: Press '1' or The complete Test Audio Sample: Press '0' ?")

take = int(input().strip())

if take == 1:

rec()

time.sleep(1.0)

path = "recorded.wav"

print ("Testing Audio : ", path)

sr,audio = read(source + path)

vector = extract\_features(audio,sr)

log\_likelihood = np.zeros(len(models))

for i in range(len(models)):

gmm = models[i]

scores = np.array(gmm.score(vector))

log\_likelihood[i] = scores.sum()

vld=abs(int(np.max(log\_likelihood)))

print(vld)

if vld<=25:

winner = np.argmax(log\_likelihood)

print(winner)

print ("\tdetected as - ", speakers[winner])

for i in accountdb:

print(i)

if i["name"] == speakers[winner]:

gui({"name":speakers[winner],"balance":i["balance"]})

else:

print("Speaker not found")

SpeakText("Speaker not found")

time.sleep(1.0)

elif take == 0:

test\_file = "testSamplePath.txt"

file\_paths = open(test\_file,'r')

print (error, total\_sample)

# Read the test directory and get the list of test audio files

for path in file\_paths:

total\_sample += 1.0

path = path.strip()

print ("Testing Audio : ", path)

sr,audio = read(source + path)

vector = extract\_features(audio,sr)

log\_likelihood = np.zeros(len(models))

for i in range(len(models)):

gmm = models[i] #checking with each model one by one

scores = np.array(gmm.score(vector))

log\_likelihood[i] = scores.sum()

checker\_name = path.split("\_")[0]

if speakers[winner] != checker\_name:

error += 1

time.sleep(1.0)

print (error, total\_sample)

accuracy = ((total\_sample - error) / total\_sample) \* 100

print ("The Accuracy Percentage for the current testing Performance with MFCC + GMM is : ", accuracy, "%")

print ("Hurrey ! Speaker identified. Mission Accomplished Successfully. ")