""" Text Extractor Project Start Here """

import os

#importing perform os operations such has filehandling and exit status

import tkinter as tk

from tkinter import \*

from tkinter.filedialog import askopenfilename

#tkinter is used to design graphical user interface in python

from PIL import ImageTk, Image

#PIL known as pillow use to open and manuplate image files

import cv2

#CV2 is name of OpenCv module which has many functions to process an image

import matplotlib.pyplot as plt

#matplotlib module used in machine learning to plots graphs and to do some mathematic computations

import matplotlib.image as mpimg

import numpy as np

#numpy module is used to process and manuplate arrays efficiently in python

import pytesseract

#pytesseract is module to process OCR (optical Character recognition)

pytesseract.pytesseract.tesseract\_cmd = 'C:/Program Files (x86)/Tesseract-OCR/tesseract'

#This is the where you have installed your pytesseract Software in Windows or linux

src\_path = "D:\\BOOKS\\ENGINEERING\\CSE\\Others\\college\_project\\"

#temprary source path to write temprary files while processing the pattern recognition

language = "eng"

#Choosing Default Language to be English

labelfont = ('times', 20, 'bold')

#definig fonts properties name,size,type to show extracted test in tkinter window

def get\_string(img\_path): #function that will extract text from image

"""get\_string(img\_path)->This function will open file img\_path and will extract the text from the file and return it as a result """

img = cv2.imread(img\_path)

#opening given image using OpenCv modules function imread

img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

#processing colors and their saturation in OpenCv

kernel = np.ones((1, 1), np.uint8)

#createing an np array of ones named kernel to dilate the image

img = cv2.dilate(img, kernel, iterations=1)

#dilate function increases the white noise in image and increases the size of pattern to extract properly

img = cv2.erode(img, kernel, iterations=1)

#erode is opposite of dilate it decreases the white noise from image and increases the pattern size so the ocr eaisly can match the pattern

cv2.imwrite(src\_path + "removed\_noise.png", img)

#writing the image which is noise removed in soruce path

cv2.imwrite(src\_path + "thres.png", img)

#writing the image which is dilate removed noise in location source path

result = pytesseract.image\_to\_string(Image.open(src\_path + "thres.png"),lang=language)

#using pytesseract extracting text from image and storing in result variable

os.system('rm ./thres.png')

#removing temparory file

os.system('rm ./removed\_noise.png')

#removing temparory file

return result

#returning extract text to tkinter window

def Application():

#function for dialogue box to open image file

"""Application()-> This function is used to open a dialogue box to select or upload an image file to extract text and storing the path of the file into global variable name for future use. """

global name

#global variable name

name = askopenfilename()

#storing address of image into name variable via dialoge box of tkinter module

def App\_Photo():

#Function to show given image at output

"""App\_Photo()-> This function will show the given image at output screen using matplotlib function plt."""

img=mpimg.imread(name)

#reading image from given path and storing in img object using matplotlib imread function

imgplot = plt.imshow(img)

#plotting Image on output screen

plt.show()

#matplotlib module function plt.show() used to show image plots on screen

def App\_Img():

#GUI window to show result

"""App\_Photo()->This function will create a result window to show extracted text at the center of the window or will display Error if something is wrong"""

global name

#using global name variable which stores address of file

n = Tk()

#creating a root window n

n.configure(background="#666666")

#setting background color of root window to gray

n.title("RESULT")

#Setting Title of root Window as RESULT

n.wm\_minsize(250,250)

#Setting fixing minimum size of root window as 250 by 250 pixels

global ch1,ch2,language

#accessing global variable ch1, ch2 and language

if ( ch1.get() and ch2.get() ) or ( not ch1.get() and not ch2.get() ) :

#condition to check wheter a user has selected both languages or does not select any of them (english or hindi)

f = Frame(n)

#wil create a frame f in root window

l = Label(f,text="Error:Choose one\nEnglish or Hindi \n",font=labelfont,bg="#666666",fg="yellow")

#will create a Label inside Frame f to display Error message

l.pack()

#attaching the lable in Frame

f.pack()

#attaching the Frame f into root Window

exit\_button = Button(n,text='Exit',command=n.destroy,width=16,height=2,bg='#666666',fg='#FFFFFF',relief=RIDGE,bd=5)

#will create a exit button in root window

exit\_button.pack(side='bottom')

#will attach exit button inside the root window at bottom of root window

n.mainloop()

#will display the root window on output screen

elif ch1.get() :

#condition to check if selected language is english

language = "eng"

#setting language to english to extract the text

else :

language = "hin"

#setting language to hindi to texract the text

var = get\_string(name)

#calling get\_string function which will return extracted text from image which will be stored in var variable

nw = Label(n,text=var,bg='#666666',fg='yellow',font=labelfont)

#createing a Lable to display the resulted text inside root Window

nw.pack(padx=50,pady=50)

#using padding to adjust text inside the root window and attaching to root window

exit\_button = Button(n,text='Exit',command=n.destroy,width=16,height=2,bg='#666666',fg='#FFFFFF',relief=RIDGE,bd=5)

#Exit button to close the window

exit\_button.pack(side='bottom')

#attaching exit button to the root window

center(n)

#open window at the center of the laptop screen

n.mainloop()

#display the window

def center(toplevel):

"""Function to adjust a tkinter window at center position of the screen """

toplevel.update\_idletasks()

#getting infromation about top level task

w = toplevel.winfo\_screenwidth()

#grabing width of screen

h = toplevel.winfo\_screenheight()

#grabing height of screen

size = tuple(int(\_) for \_ in toplevel.geometry().split('+')[0].split('x'))

#calculating size of the window

x = w/2 - size[0]/2

#horizontal center

y = h/2 - size[1]/2

#vertical center

toplevel.geometry("%dx%d+%d+%d" % (size + (x, y)))

#attach root window at the center of screen

def main():

#This is Main Window of Project

"""main()->Main Gui Window Function to run the entire project """

global ch1,ch2,name

#to access global variable ch1, ch2 and name

root = Tk()

#Root window as root

root.configure(background="#666666")

#background of root window will be gray

root.wm\_title('TextExtractor')

#title of window will be TextExtractor

root.wm\_minsize(300,300)

#minimum size of window

root.wm\_maxsize(300,300)

#maximum size of window

w = Frame(root,bg="#666666",padx='20',pady='50')

#frame w in root window

input\_button = Button(w,text='Input Image',command=Application,width=16,height=2,bg='#666666',fg='#FFFFFF',relief=RIDGE,bd=5)

#button to take image input from user

output\_button = Button(w,text='Show Output',command=App\_Img,width=16,height=2,bg='#666666',fg='#FFFFFF',relief=RIDGE,bd=5) #App\_Img

#output button to show result

ch1 = IntVar()

#dynamic variable of tkinter type integer

ch2 = IntVar()

#dynamic variable of tkinter type integer

fr = Frame(w)

#Frame fr inside window w

Checkbutton(fr,text='English',variable=ch1,bg='#666666',fg='#AAAAAA').grid(row=0,column=1,columnspan=2)

#check button 1 for english language

Checkbutton(fr,text='Hindi',variable=ch2,bg='#666666',fg='#AAAAAA').grid(row=0,column=3,columnspan=2)

#check button 2 for hindi language

input\_button.pack()

#attach input button to fr frame

fr.pack()

#attach fr frame into w window

output\_button.pack()

#atach output button to w window

buton = Button(w,text='Show Image',command=App\_Photo,width=16,height=2,bg='#666666',fg='#FFFFFF',relief=RIDGE,bd=5)

#button to display image

buton.pack()

#attach show button to w window

exit\_button = Button(w,text='Exit',command=root.destroy,width=16,height=2,bg='#666666',fg='#FFFFFF',relief=RIDGE,bd=5)

#exit button to Exit the Application

exit\_button.pack()

#attach exit button to w window

w.pack()

#attach w Frame into root windwo

center(root)

#will display window at exact center of any o/p screen

root.mainloop()

#to show main window

if \_\_name\_\_ == "\_\_main\_\_" :

"""Text Extractor Project"""

print("Welcome to Text Extraction Application".center(540,'\*'))

main() #calling main Function

print("\n\n\n\n")

print("Thanks for using Text Extraction Application".center(540,'\*'))

""" Text Extract Project Ends Here """

"""

Created by :

Sachin Yadav 14EARCS094

Samunder Arora 14EARCS097

Vijay Jangid 14EARCS119

Tanvi Goswami 14EARCS117

Charulata 14EARCS033

"""