**AI in Smart Cities**

**INT-404**

**TOPIC: [ P-02 ]**

Improving waste management through real-time

monitoring and data analysis. [CODE]

**SUBMITTED BY:**

|  |  |  |
| --- | --- | --- |
| **NAME** | **ROLL NO** | **REG NO** |
| Prince kumar | K21WYA02 | 12112403 |
| Amit gupta | K21WYA27 | 12112447 |
| Cheeraboina  jayaraju | K21WYB52 | 12109258 |

**SUBMITTED TO:**

Dr.Prateek Agrawal

#importing libaries

from cvzone.ClassificationModule import Classifier

import cvzone

import cv2

from tkinter import \*

from PIL import Image,ImageTk

from tkinter import filedialog

import os

'''

def showimg():

fln=filedialog.askopenfilename(initialdir=os.getcwd(),title="select img file ",

filetypes=(("JPG file","\*.jpg"),("PNG file ","\*.png"),("All Files","\*.\*")))

img =Image.open(fln)

img.thumbnail((350,350))

img=ImageTk.PhotoImage(img)

lbl.configure(image=img)

lbl.image=img

return fln

'''

#adding type of wastein imgwaste array

imgwaste=[]

pathfolderwaste="code/wastetypee"

pathlist=os.listdir(pathfolderwaste)

print(pathlist)

for i in pathlist:

imgwaste.append(cv2.imread(os.path.join(pathfolderwaste,i),cv2.IMREAD\_UNCHANGED))

#adding types of bin in imgbin array

imgbin=[]

pathfolderbin="code/Bins"

pathlist2=os.listdir(pathfolderbin)

print(pathlist2)

for j in pathlist2:

imgbin.append(cv2.imread(os.path.join(pathfolderbin,j),cv2.IMREAD\_UNCHANGED))

def xx():

#for opening the file and selecting the image

fln=filedialog.askopenfilename(initialdir=os.getcwd(),title="select img file ",

filetypes=(("JPG file","\*.jpg"),("PNG file ","\*.png"),("All Files","\*.\*")))

#cap = cv2.VideoCapture(1)#for ext camera

#loding the trained module

maskClassifier = Classifier('code/model/keras\_model.h5', 'code/model/labels.txt')

#seting background image

imgbackground=cv2.imread('code/pic/cc.jpg')

#copy relative path and call the fun by select image

#reading the image

img2=cv2.imread(fln)

#geting prediction which type of waste

predection = maskClassifier.getPrediction(img2)

print(predection)

classid=predection[1]

print(classid)

if classid!=0:

imgbackground=cvzone.overlayPNG(imgbackground,imgwaste[classid],(860,80,20))

imgbackground=cvzone.overlayPNG(imgbackground,imgbin[classid],(860,300))

elif classid==0:

imgbackground=cvzone.overlayPNG(imgbackground,imgwaste[0],(860,90))

imgbackground=cvzone.overlayPNG(imgbackground,imgbin[0],(560,300))

imgresize=cv2.resize(img2,(454, 340))

imgbackground[148:148+340, 159:159 + 454]=imgresize

img3=cv2.resize(imgbackground,(950,650))

#displaying background image

cv2.imshow("bgmi",img3)

cv2.waitKey(0)

#using tkinter making a search frame

root=Tk()

root.title("WASTE CLASSIFY 👇 ")

root.geometry("400x250")

frm=Frame(root)

frm.pack(side=BOTTOM,padx=35,pady=25)

lbl=Label(root)

lbl.pack()

#btn=Button(frm,text="brows img",command=showimg)

Canvas=Canvas(root,width= 800,height= 560)

#btn.pack(side=LEFT)

#calling the main fun xx by clicking the button

btn2=Button(frm,text="SEARCH 🔍",border=5, bg='green',fg='white',font='sans 16 bold',command=xx)

btn2.pack(side=RIGHT)

image=ImageTk.PhotoImage(Image.open("C:\\Users\\DELL\\OneDrive\\Desktop\\AI PROJ\\code\\pic\\OIP (1).jpg"))

Canvas.create\_image(200,90,image=image)

Canvas.pack()

root.mainloop()