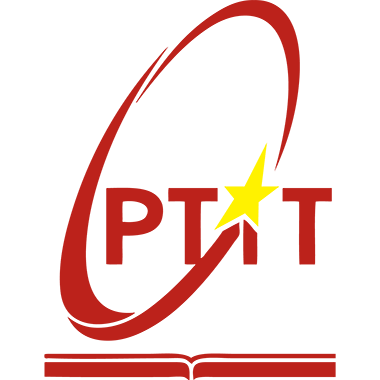
# Description: khung doi**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN 2**

🞠🙢🕮🙠🞠



**THỰC HÀNH CHUYÊN SÂU**

**ĐỀ TÀI: ỨNG DỤNG OPENCV ĐỂ XÂY DỰNG CHƯƠNG TRÌNH PHÁT HIỆN HÌNH ẢNH**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn**  **Sinh viên thực hiện**  **Mã số sinh viên**  **Lớp** | **: Th.s NGUYỄN VĂN SÁU**  **: NGUYỄN LÊ HOÀNG THANH**  **: N17DCPT058**  **: D17CQPU01-N** |

**TP. HỒ CHÍ MINH, 08/2021**

I.Tên đề tài:

Ứng dụng Opencv để xây dựng chương trình phát hiện hình ảnh

II.Mục tiêu đề tài:

Phát hiện được trong video hoặc camera hình ảnh mà người dùng muốn tìm kiếm

III.Khái niệm

Feature(điểm đặc trưng): là khu vực hình ảnh độc đáo và dễ nhận ra như điểm, cạnh, màu sắc, góc.

Feature Detection(Nhận dạng điểm đặc trưng): Tìm các điểm đặc trưng có trong hình ảnh với đầu vào là hình ảnh và đầu ra là các tọa độ pixel điểm đặc trưng của hình ảnh.

Feature Matching(Đối sánh điểm đặc trưng): So sánh các điểm đặc trưng của các hình ảnh để tìm hình ảnh tương đồng

Image Classification(Phân loại hình ảnh): là nhiệm vụ gán một ảnh đầu vào, một label từ một nhóm danh mục cố định. Đây là một trong những vấn đề cốt lõi của Computer Vision.

IV.Nội dung

**Thư viện sử dụng**

import cv2  
import numpy as np  
import os  
import shutil

**Thêm hình ảnh**  
path = 'KhoAnh'  
#orb= cv2.ORB\_create(nfeatures=1000)  
brisk = cv2.BRISK\_create()

images = []  
classNames = []  
print("Them anh vao kho?(Y/N):")  
i= input()  
if i == 'Y':  
 while True:  
 print("Duong dan file:")  
 x = input()  
 shutil.copy2(x, path)  
 print("Tiep tuc them file(Y/N):")  
 y=input()  
 if y == 'N':  
 break  
myList = os.listdir(path)

**Phân loại hình ảnh**  
for cl in myList:  
 imgCur= cv2.imread(f'{path}/{cl}',0)  
 images.append(imgCur)  
 classNames.append(os.path.splitext(cl)[0])

**Nhận dạng điểm đặc trưng của hình ảnh cần nhận dạng**

def findDes(images):  
 desList=[]  
 for img in images:  
 #kp,des = orb.detectAndCompute(img,None)  
 kp, des = brisk.detectAndCompute(img, None)  
 desList.append(des)  
 return desList

**Đối sánh điểm đặc trưng từ video hoặc camera để tìm hình ảnh cần nhận dạng**

def findID(img, desList,thres=30):  
 #kp2,des2=orb.detectAndCompute(img,None)  
 kp2, des2 = brisk.detectAndCompute(img, None)  
 bf = cv2.BFMatcher()  
 matchList=[]  
 finalVal =-1  
 try:  
 for des in desList:  
 matches = bf.knnMatch(des,des2,k=2)  
 good=[]  
 for m,n in matches:  
 if m.distance <0.75\*n.distance:  
 good.append([m])  
 matchList.append(len(good))  
 except:  
 pass  
 #print(max(matchList))  
 if len(matchList)!=0:  
 if max(matchList) > thres:  
 finalVal= matchList.index(max(matchList))  
 return finalVal  
**Mở camera hoặc video**

c=input('Camera: 0\nVideo : 1\n')  
if c == '0' :  
 cap= cv2.VideoCapture('https://192.168.50.120:8080/video')  
 if cap.isOpened()!= True :  
 cap = cv2.VideoCapture(0)  
if c == '1' :  
 x = input('Duong dan file:')  
 cap = cv2.VideoCapture(x)

**Chạy chương trình và hiển thị kết quả**

desList = findDes(images)  
while True:  
 success, img2 = cap.read()  
 imgOri= img2.copy()  
 img2=cv2.cvtColor(img2,cv2.COLOR\_BGR2GRAY)  
 id= findID(img2,desList)  
 if id != -1:  
 cv2.putText(imgOri,classNames[id],(50,50),cv2.FONT\_HERSHEY\_COMPLEX,1,(0,0,255),2)  
  
 cv2.imshow('img2',imgOri)  
 if cv2.waitKey(1) & 0xFF == ord('q'):  
 exit(0)  
cap.release()  
cv2.destroyAllWindows()

V.Kết quả đề tài

Đạt được:

+Nhận dạng được hình ảnh mà người dùng tải lên

+Người dùng có thể chọn dữ liệu hình ảnh phù hợp với nhu cầu

+Kết quả hiển thị tức thời trên màn hình

Hạn chế:

+Nhận dạng tốt với hình ảnh dạng 2D, chưa tốt với ảnh dạng 3D

+Nhận dạng từ video chậm