✅تمرین اول( کرنر)

استاد: آقای دکتر اسلامی  
نام دانشجو: مبینا دین پرست

در پایتون و یا متلب برنامه ایی را بنویسید که کرنر را پیدا کند؟  
  
  
برای پیدا کردن کرنر در تصویر یک الگوریتم معمول استفاده می‌شود به نام هاریس که به کمک مشتقات گرادیان تصویر، نقاطی از تصویر را که تفاوت بیشتری نسبت به نقاط هم‌جوار آن‌ها دارند، به عنوان کرنر تشخیص می‌دهد. در زیر الگوریتم ایجاد قالبی برای پیدا کردن کرنر در تصاویر داده شده است:  
  
زبان متلب:  
  
% ابتدا تصویر ورودی را بارگیری کنید   
I = imread('image.jpg');  
  
% تبدیل تصویر از RGB به سیاه‌سفید  
I = rgb2gray(I);  
  
% پیدا کردن مشتقات گرادیان افقی و عمودی  
[Ix,Iy] = imgradientxy(I);  
  
% نمایش مشتقات گرادیان  
figure, imshow(Ix);  
title('مشتق گرادیان افقی');  
figure, imshow(Iy);  
title('مشتق گرادیان عمودی');  
  
% پیدا کردن ماتریس هسین  
Ix2 = Ix .^ 2;  
Iy2 = Iy .^ 2;  
Ixy = Ix.\*Iy;  
sigma = 2;  
gx = fspecial('gaussian',ceil(6\*sigma), sigma);  
gx = gx / sum(gx(:));  
Ix2 = imfilter(Ix2, gx, 'same', 'conv');  
Iy2 = imfilter(Iy2, gx, 'same', 'conv');  
Ixy = imfilter(Ixy, gx, 'same', 'conv');  
cim = (Ix2.\*Iy2 - Ixy.^2) - 0.04\*(Ix2+Iy2).^2;  
  
% نمایش تصویر تشدید کرنر  
figure, imshow(cim, []);  
title('تصویر تشدید کرنر');  
  
% پیدا کردن محل کرنر با بیشینه‌یابی لوکال  
window\_size = 5;  
R = ordfilt2(cim, window\_size^2, ones(window\_size));   
cim = (cim==R)&(cim>0.01\*max(cim(:)));  
[x,y] = find(cim);  
  
% نمایش نقاط کرنر  
imshow(I), hold on, plot(y,x,'or'), hold off;  
  
  
زبان پایتون:  
  
import cv2  
import numpy as np  
  
# ابتدا تصویر ورودی را بارگیری کنید   
img = cv2.imread('image.jpg')  
  
# تبدیل تصویر از RGB به سیاه‌وسفید  
gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
  
# پیدا کردن مشتقات گرادیان افقی و عمودی  
Ix, Iy = cv2.spatialGradient(gray)  
  
# نمایش تصاویر مشتق گرادیان  
cv2.imshow('مشتق گرادیان افقی', Ix)  
cv2.imshow('مشتق گرادیان عمودی', Iy)  
  
# پیدا کردن ماتریس هسین  
Ix2 = Ix  2  
Iy2 = Iy  2  
Ixy = Ix \* Iy  
sigma = 2  
kernel\_size = int(6\*sigma)  
if kernel\_size % 2 == 0:  
    kernel\_size += 1  
gx = cv2.getGaussianKernel(kernel\_size, sigma)  
Ix2 = cv2.filter2D(Ix2, -1, gx, borderType=cv2.BORDER\_CONSTANT)  
Iy2 = cv2.filter2D(Iy2, -1, gx, borderType=cv2.BORDER\_CONSTANT)  
Ixy = cv2.filter2D(Ixy, -1, gx, borderType=cv2.BORDER\_CONSTANT)  
cim = (Ix2\*Iy2 - Ixy2) - 0.04\*(Ix2+Iy2)2  
  
# نمایش تصویر تشدید کرنر  
cv2.imshow('تصویر تشدید کرنر', cim)  
  
# پیدا کردن محل کرنر با بیشینه‌یابی لوکال  
window\_size = 5  
local\_max = cv2.dilate(cim, np.ones((window\_size,window\_size)))  
cim = (cim==local\_max)&(cim>0.01\*np.max(cim))  
x,y = np.nonzero(cim)  
  
# نمایش نقاط کرنر  
cv2.imshow('نقاط کرنر', img)  
for cx,cy in zip(y,x):  
    cv2.circle(img, (cx,cy), 2, (0,0,255), -1)  
cv2.imshow('نقاط کرنر', img)  
  
cv2.waitKey(0)