**PYTHON FOR VISION TECHNIQUES**

**WEEK - 3**

**Name:** Vatsal Panchal

**Registration number:** 18BEC1125

**CANNY EDGE DETECTOR**

import cv2

import matplotlib.pyplot as plt

import numpy as np

import matplotlib.image as mpimg

a = cv2.imread('C:/Users/WELCOME/Downloads/cameraman.png')

th1=10

th2=3\*th1

d=2

###a\_gauss=cv2.GaussianBlur(a, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

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

a\_edge = cv2.Canny(img, th1, th2)

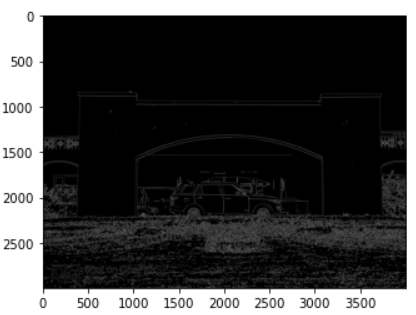
plt.figure(1)

plt.imshow(a\_edge, cmap='gray')

a\_gauss[a\_edge != 0] = (0, 255, 0)

plt.figure(2)

plt.imshow(cv2.cvtColor(a\_gauss, cv2.COLOR\_BGR2RGB))

**HARRIS CORNER DETECTOR**

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

image = cv2.imread('C:/Users/WELCOME/Downloads/cameraman.png')

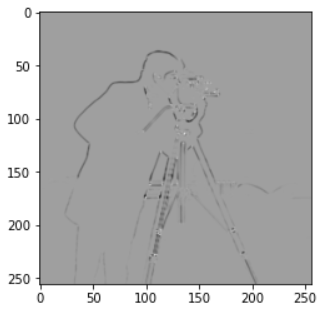
gray=cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

gray = np.float32(gray)

dst = cv2.cornerHarris(gray,2,3,0.11)

#image[dst = 0] = (0, 255, 0)

plt.imshow(dst,cmap='gray')



**HOUGH TRANSFORM – LINE DETECTION**

import cv2

import numpy as np

import matplotlib.pyplot as plt

a = cv2.imread('C:/Users/WELCOME/Desktop/New folder/line.png')

d=2

a\_gauss=cv2.GaussianBlur(a, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

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

edges = cv2.Canny(img,50,150,apertureSize = 3)

plt.figure(1)

plt.imshow(cv2.cvtColor(edges, cv2.COLOR\_BGR2RGB))

lines = cv2.HoughLines(edges,1,np.pi/180,100)

for rho,theta in lines[0]:

a = np.cos(theta)

b = np.sin(theta)

x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b))

y1 = int(y0 + 1000\*(a))

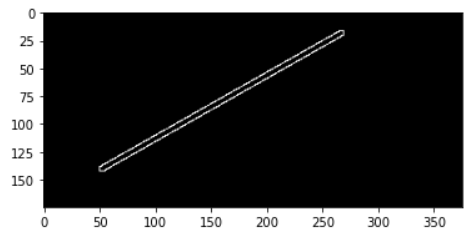
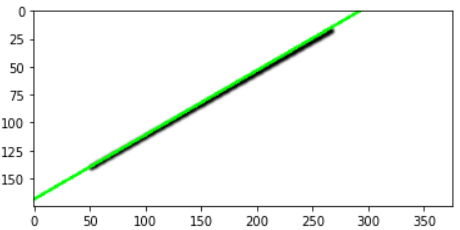
x2 = int(x0 - 1000\*(-b))

y2 = int(y0 - 1000\*(a))

cv2.line(a\_gauss,(x1,y1),(x2,y2),(0,255,0),2)

plt.figure(2)

plt.imshow(cv2.cvtColor(a\_gauss, cv2.COLOR\_BGR2RGB))

**HOUGH TRANSFORM – CIRCLE DETECTION**

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('…')

d=2

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

gimg = cv2.GaussianBlur(gimg, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

#cimg = cv2.cvtColor(img,cv2.COLOR\_GRAY2BGR)

circles = cv2.HoughCircles(gimg,cv2.HOUGH\_GRADIENT,1,20,param1=50,param2=30,minRadius=0,maxRadius=0)

circles = np.uint16(np.around(circles))

for i in circles[0,:]:

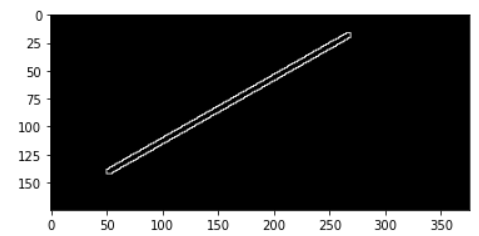
    # draw the outer circle

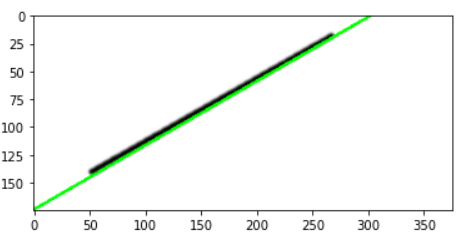
    cv2.circle(img,(i[0],i[1]),i[2],(0,255,0),2)

    # draw the center of the circle

    cv2.circle(img,(i[0],i[1]),2,(0,0,255),3)

plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))





import cv2

import numpy as np

import matplotlib.pyplot as plt

a = cv2.imread('C:/Users/WELCOME/Downloads/capture2.jpg')

d=2

a\_gauss=cv2.GaussianBlur(a, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

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

edges = cv2.Canny(img,50,150,apertureSize = 3)

plt.figure(1)

plt.imshow(cv2.cvtColor(edges, cv2.COLOR\_BGR2RGB))

lines = cv2.HoughLines(edges,1,np.pi/180,100)

for rho,theta in lines[0]:

a = np.cos(theta)

b = np.sin(theta)

x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b))

y1 = int(y0 + 1000\*(a))

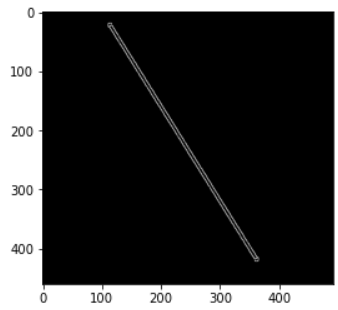
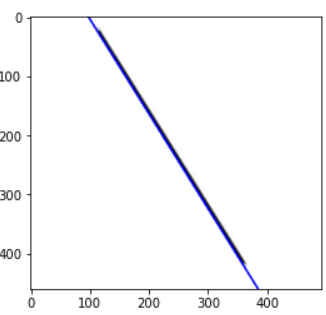
x2 = int(x0 - 1000\*(-b))

y2 = int(y0 - 1000\*(a))

cv2.line(a\_gauss,(x1,y1),(x2,y2),(255,0,0),2)

plt.figure(2)

plt.imshow(cv2.cvtColor(a\_gauss, cv2.COLOR\_BGR2RGB))

import cv2

import numpy as np

import matplotlib.pyplot as plt

a = cv2.imread('C:/Users/WELCOME/Desktop/New folder/Checkerboard.png')

d=2

a\_gauss=cv2.GaussianBlur(a, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

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

edges = cv2.Canny(img,50,150,apertureSize = 3)

plt.figure(1)

plt.imshow(cv2.cvtColor(edges, cv2.COLOR\_BGR2RGB))

lines = cv2.HoughLines(edges,1,np.pi/180,100)

for i in range (lines.shape[0]):

for rho,theta in lines[i]:

a = np.cos(theta)

b = np.sin(theta)

x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b))

y1 = int(y0 + 1000\*(a))

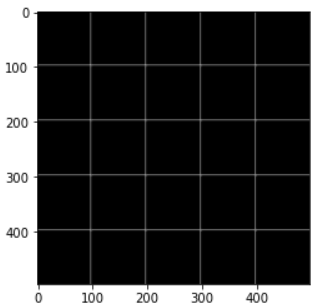
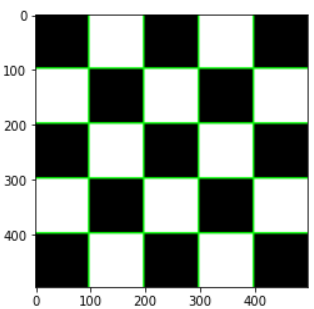
x2 = int(x0 - 1000\*(-b))

y2 = int(y0 - 1000\*(a))

cv2.line(a\_gauss,(x1,y1),(x2,y2),(0,255,0),2)

plt.figure(2)

plt.imshow(cv2.cvtColor(a\_gauss, cv2.COLOR\_BGR2RGB))

import cv2

import numpy as np

import matplotlib.pyplot as plt

a = cv2.imread('C:/Users/WELCOME/Downloads/cameraman.png')

d=2

a\_gauss=cv2.GaussianBlur(a, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

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

edges = cv2.Canny(img,50,150,apertureSize = 3)

plt.figure(1)

plt.imshow(cv2.cvtColor(edges, cv2.COLOR\_BGR2RGB))

lines = cv2.HoughLines(edges,1,np.pi/180,100)

for i in range (lines.shape[0]):

for rho,theta in lines[i]:

a = np.cos(theta)

b = np.sin(theta)

x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b))

y1 = int(y0 + 1000\*(a))

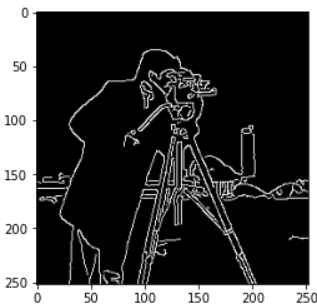
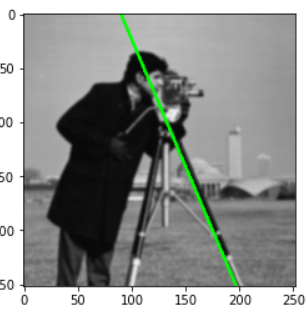
x2 = int(x0 - 1000\*(-b))

y2 = int(y0 - 1000\*(a))

cv2.line(a\_gauss,(x1,y1),(x2,y2),(0,255,0),2)

plt.figure(2)

plt.imshow(cv2.cvtColor(a\_gauss, cv2.COLOR\_BGR2RGB))

import cv2

import numpy as np

import matplotlib.pyplot as plt

a = cv2.imread('C:/Users/WELCOME/Downloads/capture.jpg')

d=2

a\_gauss=cv2.GaussianBlur(a, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

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

edges = cv2.Canny(img,50,150,apertureSize = 3)

plt.figure(1)

plt.imshow(cv2.cvtColor(edges, cv2.COLOR\_BGR2RGB))

lines = cv2.HoughLines(edges,1,np.pi/180,120)

for i in range(lines.shape[0]):

for rho,theta in lines[i]:

a = np.cos(theta)

b = np.sin(theta)

x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b))

y1 = int(y0 + 1000\*(a))

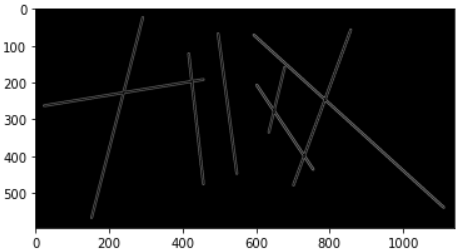
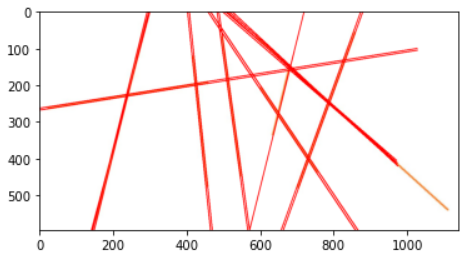
x2 = int(x0 - 1000\*(-b))

y2 = int(y0 - 1000\*(a))

cv2.line(a\_gauss,(x1,y1),(x2,y2),(0,0,255),2)

plt.figure(2)

plt.imshow(cv2.cvtColor(a\_gauss, cv2.COLOR\_BGR2RGB))

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('C:/Users/WELCOME/Desktop/New folder/Circle.jpg')

d=2

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

gimg = cv2.GaussianBlur(gimg, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

#cimg = cv2.cvtColor(img,cv2.COLOR\_GRAY2BGR)

circles = cv2.HoughCircles(gimg,cv2.HOUGH\_GRADIENT,1,20,param1=50,param2=30,minRadius=0,maxRadius=0)

circles = np.uint16(np.around(circles)

for i in circles[0,:]:

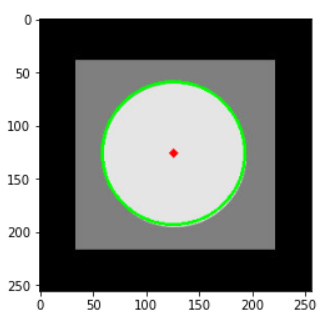
# draw the outer circle

cv2.circle(img,(i[0],i[1]),i[2],(0,255,0),2)

# draw the center of the circle

cv2.circle(img,(i[0],i[1]),2,(0,0,255),3)

plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))



import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('C:/Users/WELCOME/Downloads/capture3.jpg')

d=2

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

gimg = cv2.GaussianBlur(gimg, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

#cimg = cv2.cvtColor(img,cv2.COLOR\_GRAY2BGR)

circles = cv2.HoughCircles(gimg,cv2.HOUGH\_GRADIENT,1,180,param1=50,param2=30,minRadius=0,maxRadius=0)

circles = np.uint16(np.around(circles))

for i in circles[0,:]:

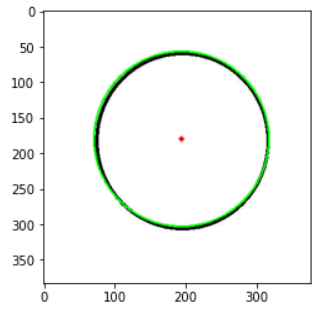
# draw the outer circle

cv2.circle(img,(i[0],i[1]),i[2],(0,255,0),2)

# draw the center of the circle

cv2.circle(img,(i[0],i[1]),2,(0,0,255),3)

plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))



import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('C:/Users/WELCOME/Downloads/capture1.jpg')

d=2

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

gimg = cv2.GaussianBlur(gimg, (2\*d+1, 2\*d+1), -1)[d:-d,d:-d]

#cimg = cv2.cvtColor(img,cv2.COLOR\_GRAY2BGR)

circles = cv2.HoughCircles(gimg,cv2.HOUGH\_GRADIENT,1,115,param1=50,param2=30,minRadius=0,maxRadius=0)

circles = np.uint16(np.around(circles))

for i in circles[0,:]:

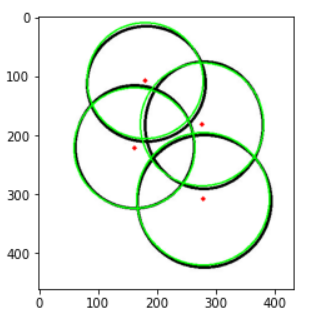
# draw the outer circle

cv2.circle(img,(i[0],i[1]),i[2],(0,255,0),2)

# draw the center of the circle

cv2.circle(img,(i[0],i[1]),2,(0,0,255),3)

plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))



**TASK 5- KMEANS CLUSTERING**

import numpy as np

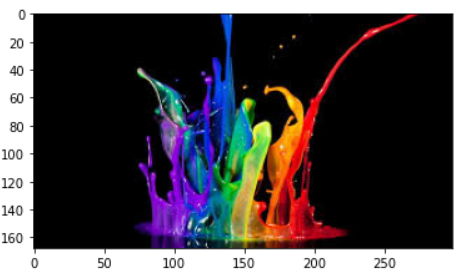
import cv2

import matplotlib.pyplot as plt

img = cv2.imread('C:/Users/WELCOME/Downloads/images.jpg')

img=np.float32(img)

plt.imshow(cv2.cvtColor(np.uint8(img),cv2.COLOR\_BGR2RGB))



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

[m, n, k]=np.shape(img)

img\_1d=img.reshape(m\*n,k)

from sklearn.cluster import KMeans

#kmeans = KMeans(n\_clusters=3, random\_state=0).fit(img\_1d)

kmeans = KMeans(n\_clusters=7).fit(img\_1d)

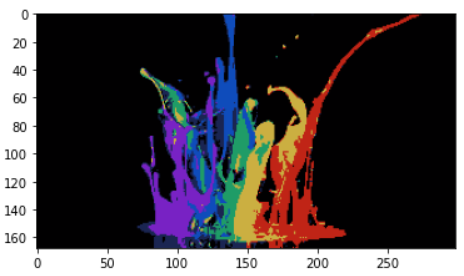
img2show = kmeans.cluster\_centers\_[kmeans.labels\_]

img\_cluster = img2show.reshape(m,n,k )

img\_cluster=np.array(img\_cluster, dtype=np.uint8)

###plt.imshow(img\_cluster)

plt.imshow(cv2.cvtColor(np.uint8(img\_cluster),cv2.COLOR\_BGR2RGB))



**OUTPUT 2-**

