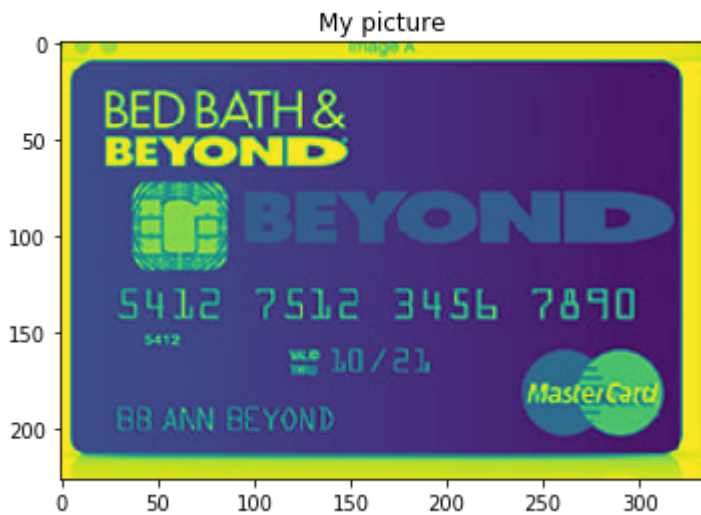


Images Edge Detection

In [2]:

```
from matplotlib import pyplot as plt
import cv2

fig=plt.figure()
img = cv2.imread(r'C:\Users\Lenovo\Desktop\images\image1.PNG')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite('edge_image1.jpg',cv2.Canny(img,200,300))
cv2.imshow('edges',cv2.imread('edge_image1.jpg'))
plt.imshow(gray)
plt.title('My picture')
plt.show()
#plt.rcParams["figure.figsize"]=(10,10)
```



In [3]:

```

from matplotlib import pyplot as plt
import numpy as np

fig=plt.figure()
img=plt.imread(r'C:\Users\Lenovo\Desktop\images\box_image.JPG')
plt.imshow(img)
grey_img=img.mean(axis=2,keepdims=True)/255.0
grey_img=np.concatenate([grey_img]*3,axis=2)
plt.imshow(grey_img)
vertical_filter=[[-1,-2,-1],[0,0,0],[1,2,1]]
horizontal_filter=[[-1,0,1],[-2,0,2],[-1,0,1]]
n,m,d=img.shape
edges_img=np.zeros_like(img)

for row in range(3,n-2):
    for col in range(3,m-2):
        local_pixels=img[row-1:row+2,col-1:col+2,0]
        vertical_transformed_pixels=vertical_filter*local_pixels
        vertical_score=vertical_transformed_pixels.sum()/4

        horizontal_transformed_pixels=horizontal_filter*local_pixels
        horizontal_score=horizontal_transformed_pixels.sum()/4

        edge_score=(vertical_score**2+horizontal_score**2)**.5

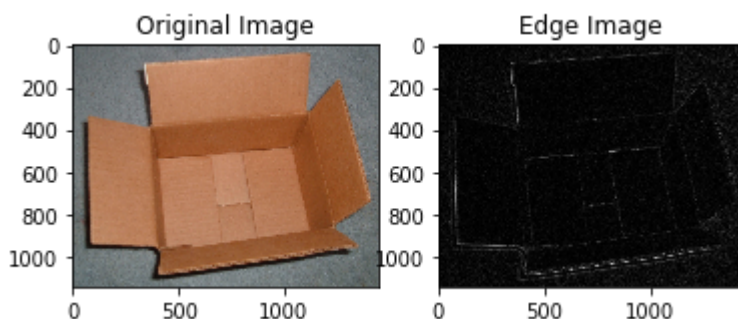
        edges_img[row,col]=[edge_score]*3
edges_img=edges_img/edges_img.max()

#plt.imshow(img)
#plt.imshow(edges_img)
plt.rcParams["figure.figsize"]=(20,8)

plt.subplot(121),plt.imshow(img)
plt.title('Original Image')
plt.subplot(122),plt.imshow(edges_img)
plt.title('Edge Image')

plt.show()

```



In [4]:

```
from matplotlib import pyplot as plt
import numpy as np

fig=plt.figure()
img=plt.imread(r'C:\Users\Lenovo\Desktop\spin an win widgets\logo.PNG')
plt.imshow(img)
grey_img=img.mean(axis=2,keepdims=True)/255.0
grey_img=np.concatenate([grey_img]*3,axis=2)
plt.imshow(grey_img)
vertical_filter=[[-1,-2,-1],[0,0,0],[2,2,1]]
horizontal_filter=[[-1,0,1],[-2,0,2],[-1,0,1]]
n,m,d=img.shape
edges_img=np.zeros_like(img)

for row in range(3,n-2):
    for col in range(3,m-2):
        local_pixels=img[row-1:row+2,col-1:col+2,0]
        vertical_transferred_pixels=vertical_filter*local_pixels
        vertical_score=vertical_transferred_pixels.sum()/4

        horizontal_transferred_pixels=horizontal_filter*local_pixels
        horizontal_score=horizontal_transferred_pixels.sum()/4

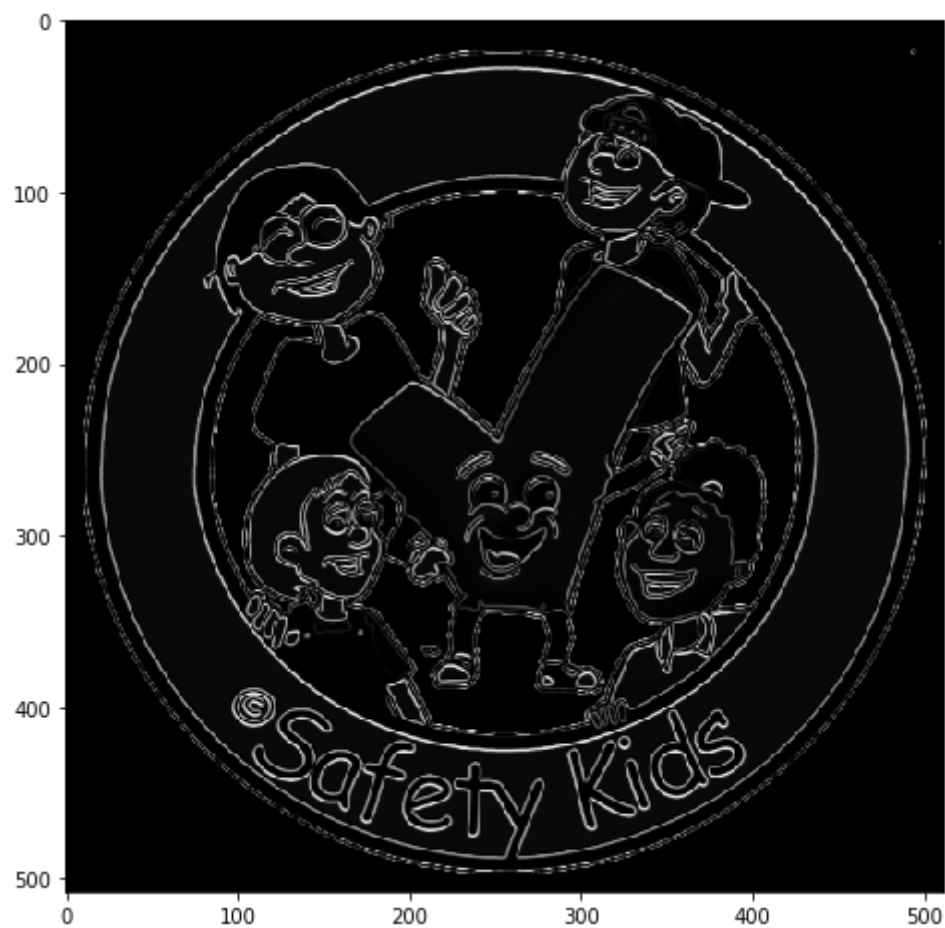
        edge_score=(vertical_score**2+horizontal_score**2)**.5

        edges_img[row,col]=[edge_score]*4
edges_img=edges_img/edges_img.max()

plt.imshow(img)
plt.imshow(edges_img)
plt.rcParams["figure.figsize"]=(10,6)

plt.subplot(121),plt.imshow(img)
plt.title('Original Image')
plt.subplot(122),plt.imshow(edges_img)
plt.title('Edge Image')

plt.show()
```



In [11]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread(r'C:\Users\Lenovo\Desktop\spin an win widgets\logo.PNG')
edges = cv2.Canny(img,200,400)

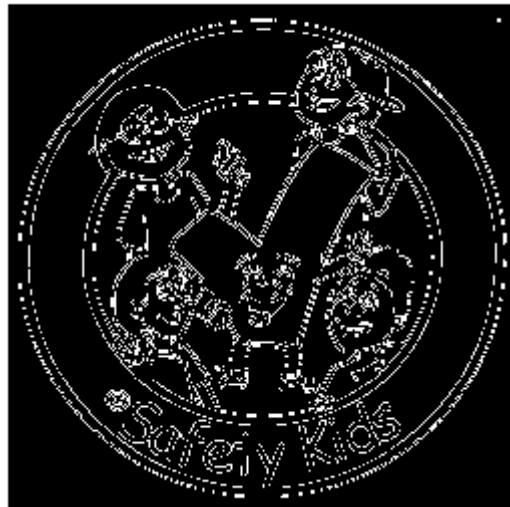
plt.subplot(121),plt.imshow(img)
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()
```

Original Image



Edge Image



In [9]:

```
from matplotlib import pyplot as plt
import numpy as np

fig=plt.figure()
img=plt.imread(r'C:\Users\Lenovo\Desktop\spin an win widgets\samsung-galaxy-m20.JPG')
plt.imshow(img)
grey_img=img.mean(axis=2,keepdims=True)/255.0
grey_img=np.concatenate([grey_img]*3,axis=2)
plt.imshow(grey_img)
vertical_filter=[[-1,-2,-1],[0,0,0],[2,2,1]]
horizontal_filter=[[-1,0,1],[-2,0,2],[-1,0,1]]
n,m,d=img.shape
edges_img=np.zeros_like(img)

for row in range(3,n-2):
    for col in range(3,m-2):
        local_pixels=img[row-1:row+2,col-1:col+2,0]
        vertical_transferred_pixels=vertical_filter*local_pixels
        vertical_score=vertical_transferred_pixels.sum()/4

        horizontal_transferred_pixels=horizontal_filter*local_pixels
        horizontal_score=horizontal_transferred_pixels.sum()/4

        edge_score=(vertical_score**2+horizontal_score**2)**.5

        edges_img[row,col]=[edge_score]*3
edges_img=edges_img/edges_img.max()

#plt.imshow(img)
plt.imshow(edges_img)
plt.rcParams["figure.figsize"]=(10,6)

#plt.subplot(121),plt.imshow(img)
#plt.title('Original Image')
#plt.subplot(122),plt.imshow(edges_img)
#plt.title('Edge Image')

#plt.show()
```



In []: