In [3]:

```
!pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\dell\anaconda3\lib
\site-packages (4.6.0.66)
Requirement already satisfied: numpy>=1.19.3 in c:\users\dell\anaconda3\lib
\site-packages (from opencv-python) (1.20.3)
```

In [4]:

```
import cv2
import matplotlib.pyplot as plt
```

In [5]:

```
img=cv2.imread("head-treasure-flower-disk-flowers-inflorescence-ray.webp")
```

In [6]:

```
plt.show
```

Out[6]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

In [7]:

```
#plt.axis("off")
plt.imshow(img)
```

Out[7]:

```
<matplotlib.image.AxesImage at 0x2435a05e070>
```

In [8]:

```
RBG_image=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.imshow(RBG_image)
```

Out[8]:

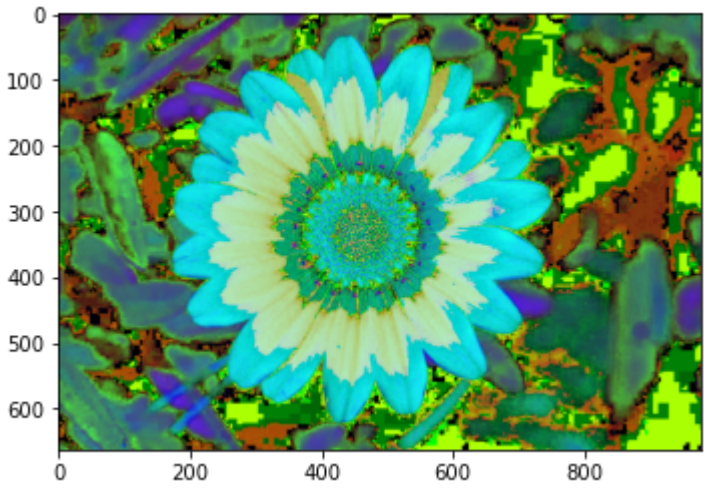
```
<matplotlib.image.AxesImage at 0x2435a150f10>
```

In [9]:

```
HSV_image=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
plt.imshow(HSV_image)
```

Out[9]:

```
<matplotlib.image.AxesImage at 0x2435ab60eb0>
```



In [10]:

```
img.shape
```

Out[10]:

```
(663, 977, 3)
```

In [11]:

```
img
```

Out[11]:

```
array([[[12, 21,  8],
        [26, 35, 22],
        [37, 46, 33],
        ...,
        [35, 79, 56],
        [34, 78, 55],
        [36, 78, 55]],

       [[26, 35, 22],
        [40, 49, 36],
        [52, 61, 48],
        ...,
        [39, 83, 60],
        [39, 83, 60],
        [40, 81, 59]],

       [[37, 46, 33],
        [52, 61, 48],
        [62, 71, 58],
        ...,
        [42, 86, 63],
        [41, 85, 62],
        [42, 84, 61]],

       ...,

       [[29, 55, 17],
        [32, 59, 20],
        [37, 63, 25],
        ...,
        [ 1,  1,  1],
        [ 1,  1,  1],
        [ 1,  1,  1]],

       [[29, 55, 17],
        [32, 59, 20],
        [37, 63, 25],
        ...,
        [ 1,  1,  1],
        [ 1,  1,  1],
        [ 1,  1,  1]],

       [[29, 55, 17],
        [32, 59, 20],
        [37, 63, 25],
        ...,
        [ 1,  1,  1],
        [ 1,  1,  1],
        [ 1,  1,  1]]], dtype=uint8)
```

In [12]:

```
img.size
```

Out[12]:

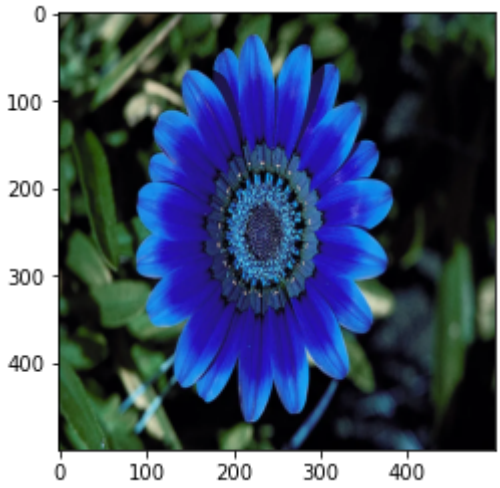
```
1943253
```

In [13]:

```
img_resized=cv2.resize(img,(500,500))
plt.imshow(img_resized)
```

Out[13]:

```
<matplotlib.image.AxesImage at 0x2435adaffa0>
```

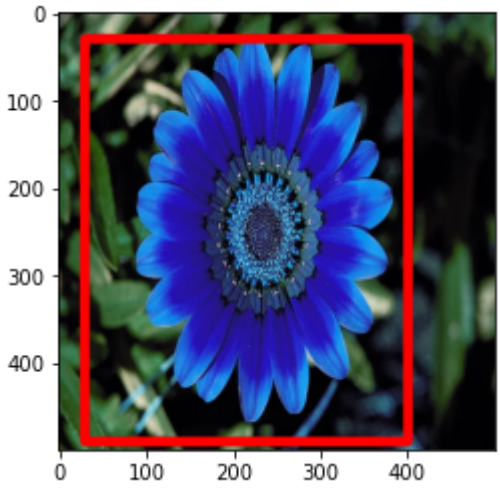


In [14]:

```
#rectangle
start_point =(30,30)
end_point=(400,490)
color=(255,0,0)
thickness=10
image=cv2.rectangle(img_resized,start_point,end_point,color,thickness)
plt.imshow(image)
```

Out[14]:

```
<matplotlib.image.AxesImage at 0x2435ae10c40>
```

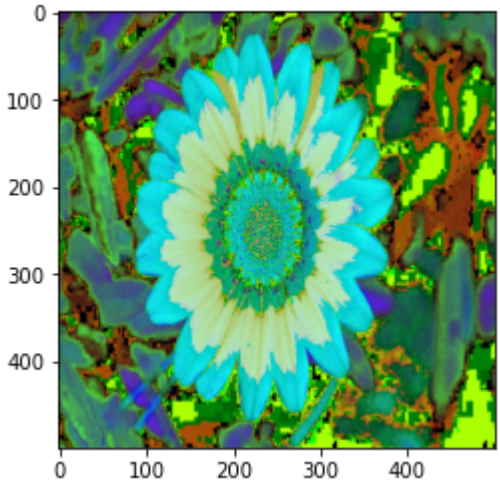


In [15]:

```
img_resized3=cv2.resize(HSV_image,(500,500))
plt.imshow(img_resized3)
```

Out[15]:

```
<matplotlib.image.AxesImage at 0x2435be436a0>
```

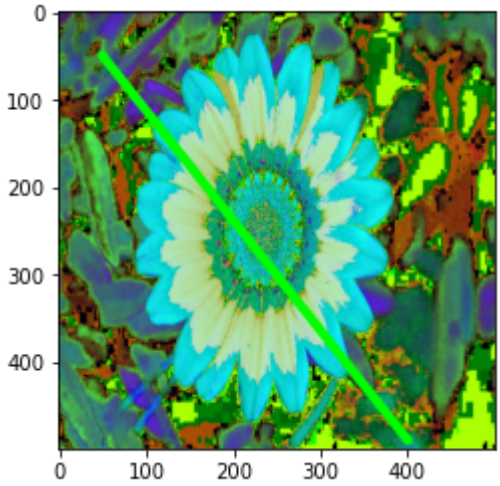


In [16]:

```
#Line
start_point =(50,50)
end_point=(400,490)
color=(0,255,0)
thickness=9
image3=cv2.line(img_resized3,start_point,end_point,color,thickness)
plt.imshow(image3)
```

Out[16]:

```
<matplotlib.image.AxesImage at 0x2435bea8220>
```



In [17]:

```
#resized the image
img_resized2=cv2.resize(RBG_image,(500,500))
plt.imshow(img_resized2)
```

Out[17]:

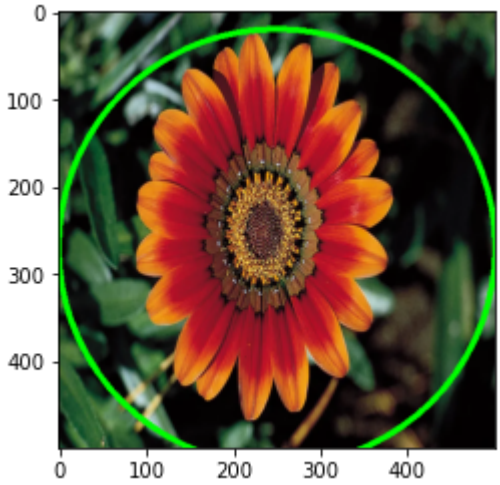
```
<matplotlib.image.AxesImage at 0x2435beff970>
```

In [18]:

```
#circle
center_corrdinates=(250,270)
radius=250
color=(0,255,0)
thickness=5
image2=cv2.circle(img_resized2,center_corrdinates,radius,color,thickness)
plt.imshow(image2)
```

Out[18]:

```
<matplotlib.image.AxesImage at 0x2435bf634c0>
```

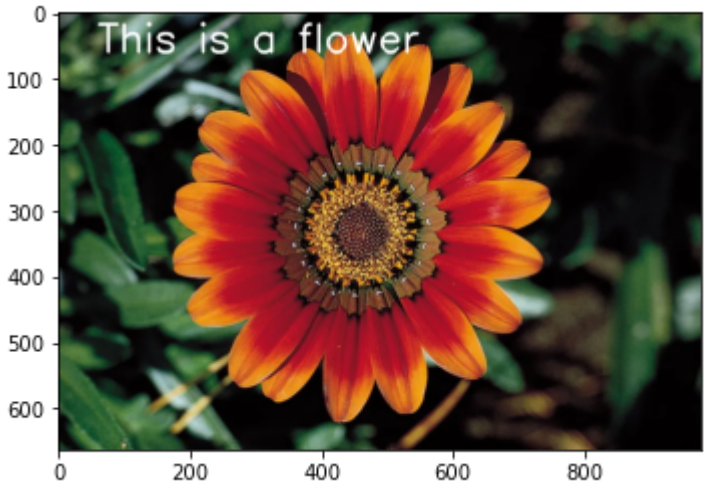


In [19]:

```
#text
font = cv2.FONT_HERSHEY_SIMPLEX
org=(60,60)
fontScale=2


color=(255,255,255)
thickness=4
image=cv2.putText(RBG_image,"This is a flower",org,font,fontScale,color,thickness)
plt.imshow(image)
```

Out[19]:

```
<matplotlib.image.AxesImage at 0x2435bfbbfd0>
```



In [20]:

```
#roation
image =cv2.transpose(RBG_image)
plt.imshow(image)
```

Out[20]:

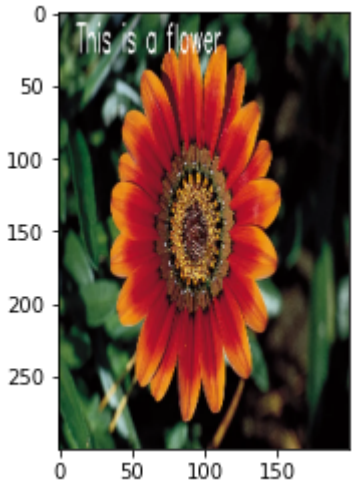
```
<matplotlib.image.AxesImage at 0x2435c3fc0d0>
```

In [21]:

```
#small size
small=cv2.resize(RBG_image,(200,300))
plt.imshow(small)
```

Out[21]:

```
<matplotlib.image.AxesImage at 0x2435c63d280>
```



In [22]:

```
s_N=cv2.resize(RBG_image,(780,540),
                 interpolation=cv2.INTER_NEAREST)
plt.imshow(s_N)
```

Out[22]:

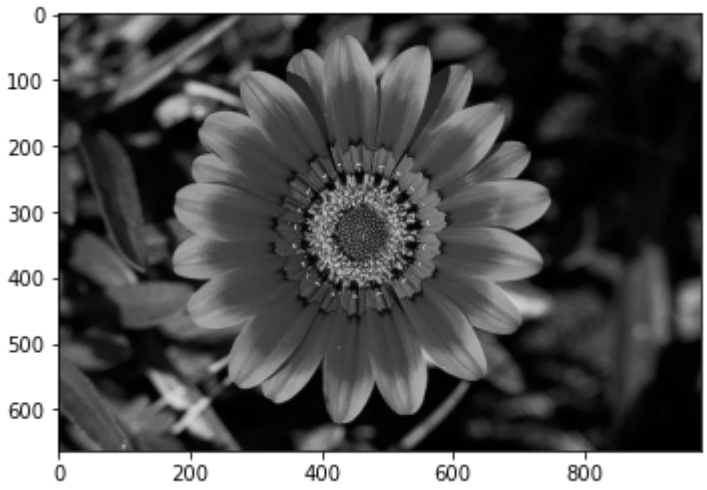
```
<matplotlib.image.AxesImage at 0x2435c7d7f70>
```

In [23]:

```
#gray_img
gray_img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
plt.imshow(gray_img,cmap="gray")
```

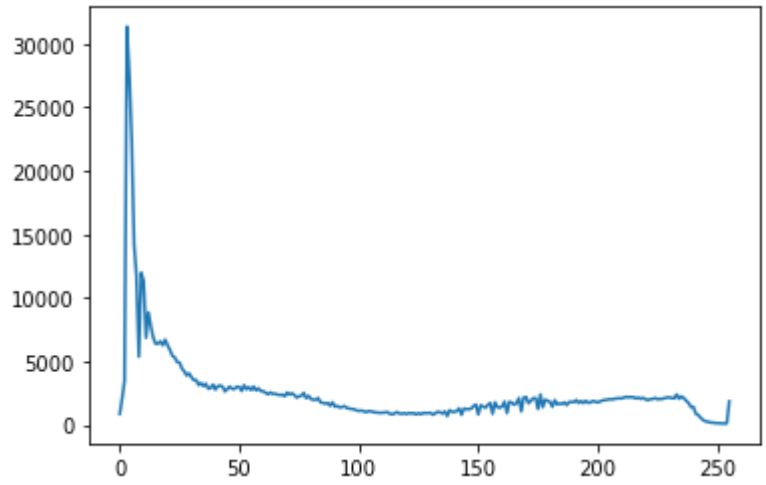
Out[23]:

```
<matplotlib.image.AxesImage at 0x2435c9920d0>
```



In [24]:

```
#histrogram plot
hist=cv2.calcHist([img],[2],None,[256],[0,256])
plt.plot(hist)
plt.show()
```

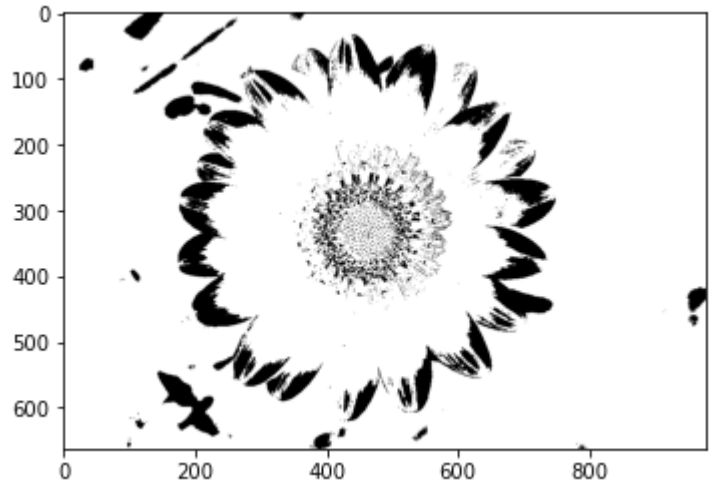


In [25]:

```
#gray img
ret,thresh2=cv2.threshold(gray_img,130,150, cv2.THRESH_BINARY_INV)
plt.imshow(thresh2,cmap="gray")
```

Out[25]:

```
<matplotlib.image.AxesImage at 0x2435ca54580>
```

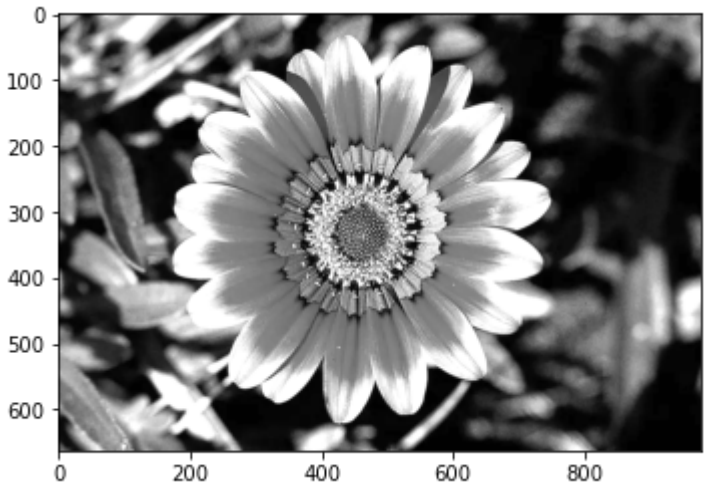


In [26]:

```
#gray img
ret,thresh3=cv2.threshold(gray_img,130,150, cv2.THRESH_TRUNC)
plt.imshow(thresh3,cmap="gray")
```

Out[26]:

```
<matplotlib.image.AxesImage at 0x2435cab49a0>
```



In [27]:

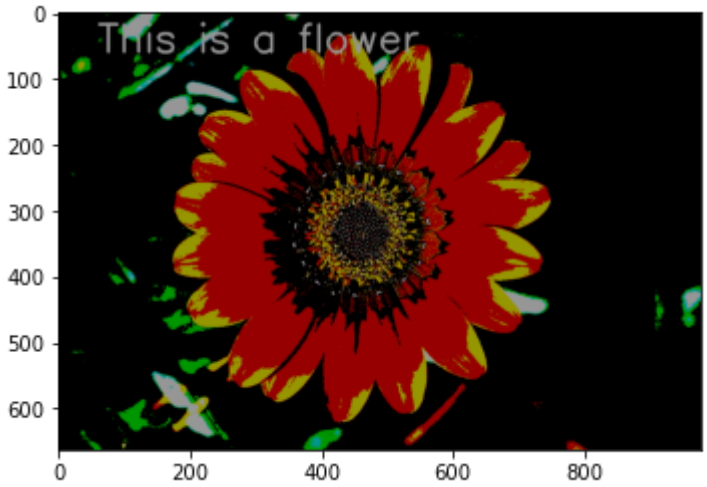
```
#RBG image
```

In [28]:

```
ret,thresh1=cv2.threshold(RBG_image,130,150, cv2.THRESH_BINARY)
plt.imshow(thresh1)
```

Out[28]:

```
<matplotlib.image.AxesImage at 0x2435dcd4190>
```



In [29]:

```
ret,thresh3=cv2.threshold(RBG_image,130,150, cv2.THRESH_TRUNC)
plt.imshow(thresh3)
```

Out[29]:

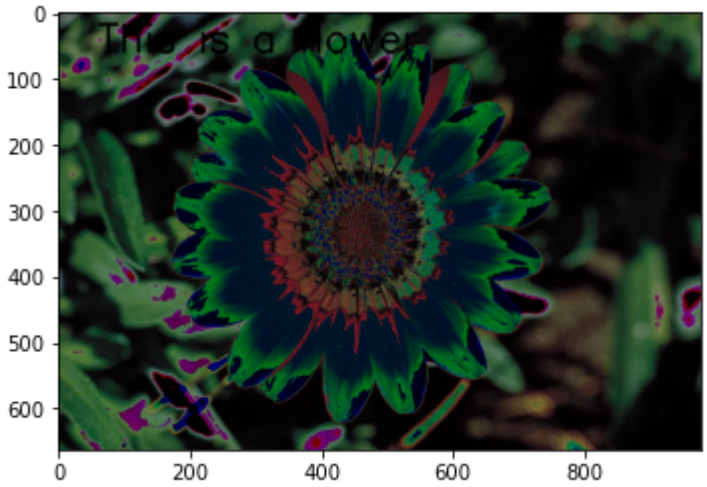
```
<matplotlib.image.AxesImage at 0x2435e1117c0>
```

In [30]:

```
ret,thresh4=cv2.threshold(RBG_image,130,150, cv2.THRESH_TOZERO_INV)
plt.imshow(thresh4,cmap="gray")
```

Out[30]:

```
<matplotlib.image.AxesImage at 0x2435e362dc0>
```

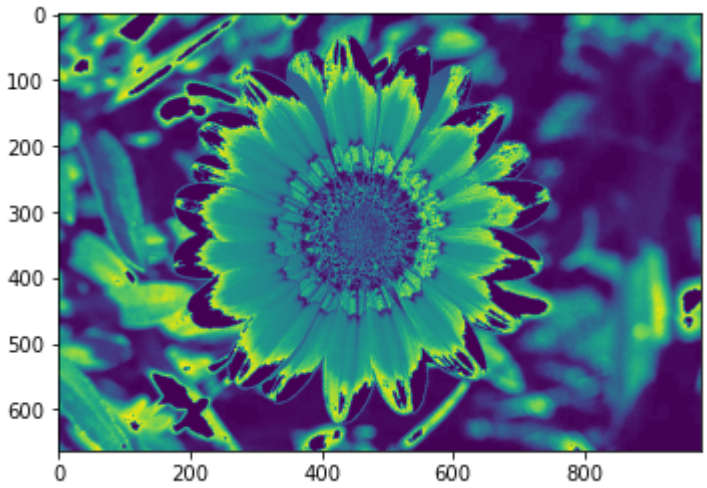


In [31]:

```
#masked gray img
masked1 = cv2.bitwise_and(gray_img,gray_img,mask=thresh2)
plt.imshow(masked1)
```

Out[31]:

```
<matplotlib.image.AxesImage at 0x2435e79d4c0>
```

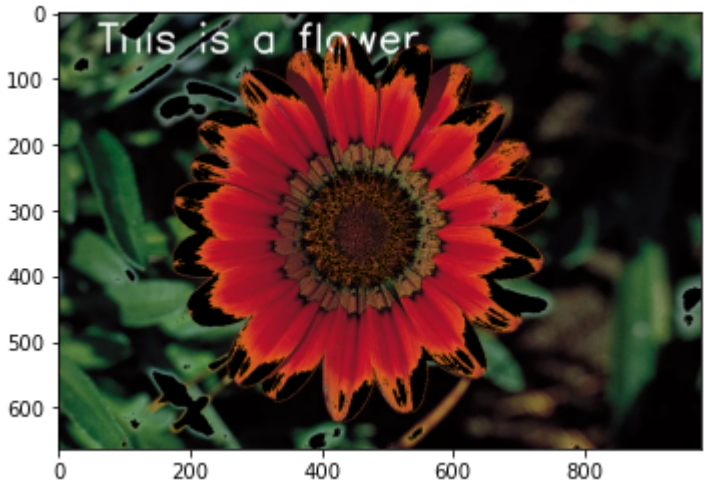


In [32]:

```
#masked RBGimage
masked2 = cv2.bitwise_and(RBG_image,RBG_image,mask=thresh2)
plt.imshow(masked2)
```

Out[32]:

```
<matplotlib.image.AxesImage at 0x2435e7ff9d0>
```



In [33]:

```
#GaussianBlur
Gaussian_Blur=cv2.GaussianBlur(RBG_image,(11,11),(0))
plt.imshow(Gaussian_Blur)
```

Out[33]:

```
<matplotlib.image.AxesImage at 0x2435ee20eb0>
```

In [34]:

```
#Median_blur
median=cv2.medianBlur(RBG_image,11)
plt.imshow(median)
```

Out[34]:

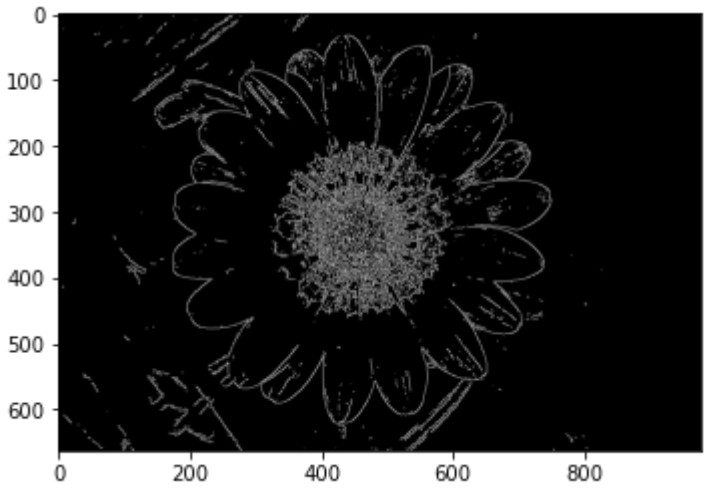
```
<matplotlib.image.AxesImage at 0x2435f26b3a0>
```

In [35]:

```
#canny grayimage
edge=cv2.Canny(gray_img,130,150)
plt.imshow(edge,cmap="gray")
```

Out[35]:

```
<matplotlib.image.AxesImage at 0x2435f4ba910>
```

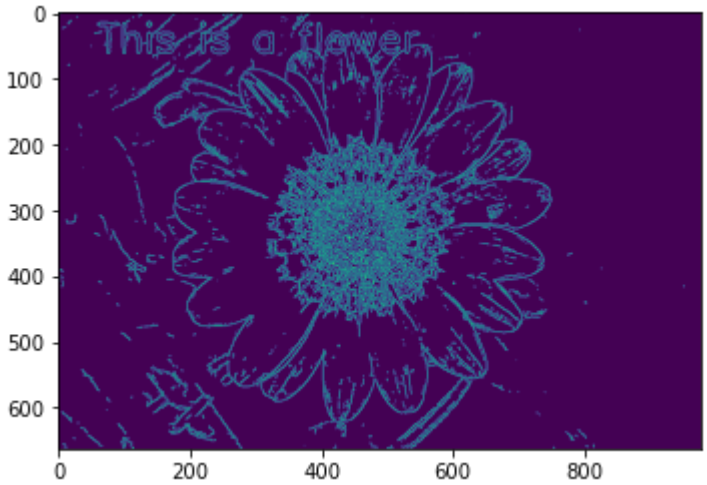


In [36]:

```
#cannyRBG_image
edge=cv2.Canny(RBG_image,130,150)
plt.imshow(edge)
```

Out[36]:

```
<matplotlib.image.AxesImage at 0x243604eddc0>
```



In [ ]:

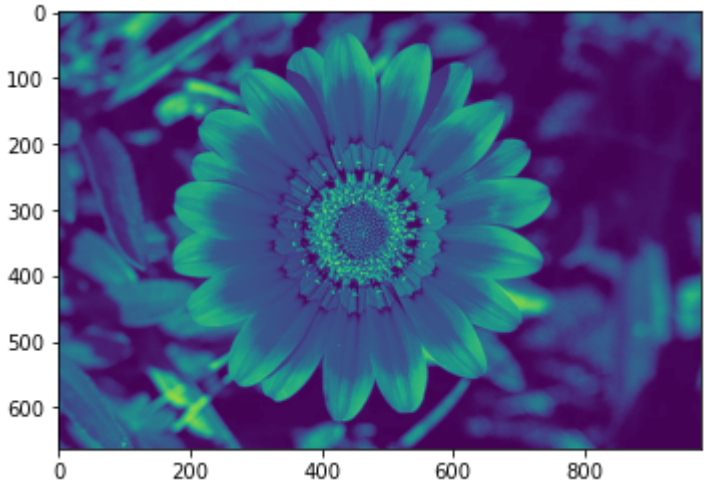
fliters

In [37]:

```
#identiy kernal
import numpy as np
filter=np.array([[0,0,0],[0,1,0],[0,0,0]])
median=cv2.filter2D(gray_img,-1,filter)
plt.imshow(median)
```

Out[37]:

```
<matplotlib.image.AxesImage at 0x2435c9439a0>
```

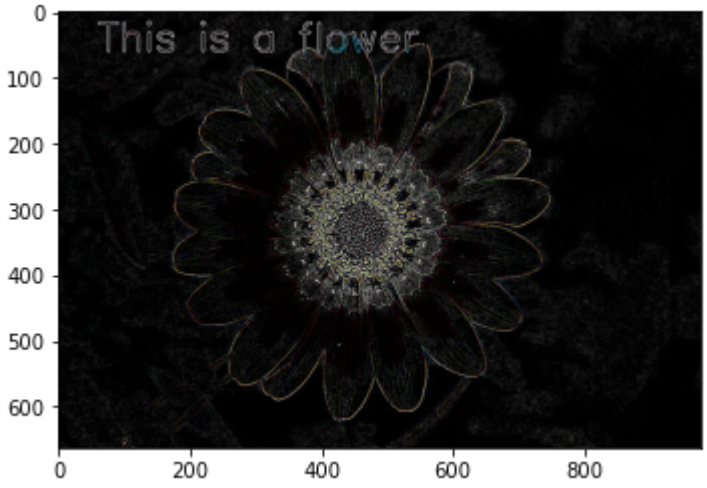


In [38]:

```
#edge detection
import numpy as np
filter=np.array([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])
median=cv2.filter2D(RBG_image,-1,filter)
plt.imshow(median)
```

Out[38]:

```
<matplotlib.image.AxesImage at 0x24360552280>
```

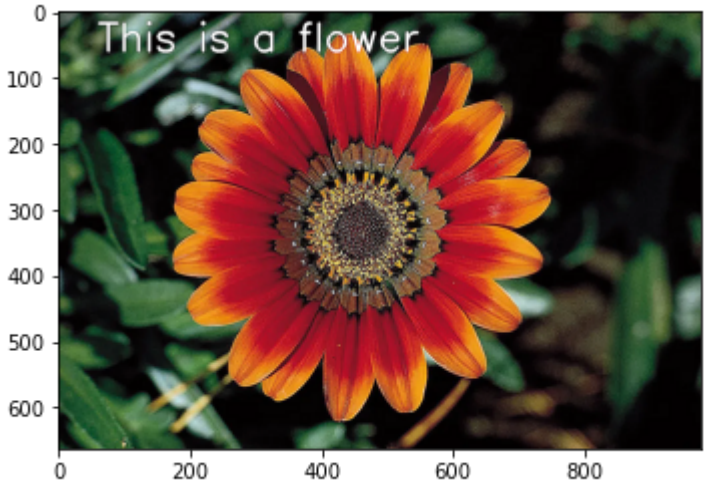


In [39]:

```
#sharpen kernal
import numpy as np
filter=np.array([[0,-1,0],[-1,5,-1],[0,-1,0]])
median=cv2.filter2D(RBG_image,-1,filter)
plt.imshow(median)
```

Out[39]:

```
<matplotlib.image.AxesImage at 0x2435f07d700>
```

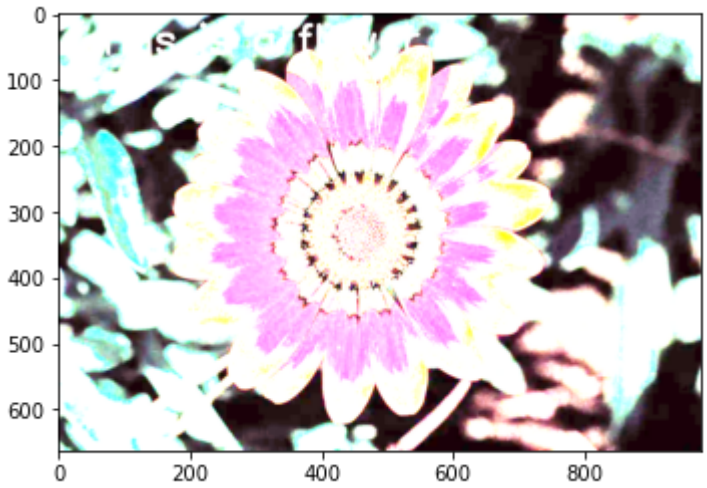


In [40]:

```
#box blur
import numpy as np
filter=np.array([[1,1,1],[1,1,1],[1,1,1]])
median=cv2.filter2D(RBG_image,-1,filter)
plt.imshow(median)
```

Out[40]:

```
<matplotlib.image.AxesImage at 0x2435f0dfb50>
```

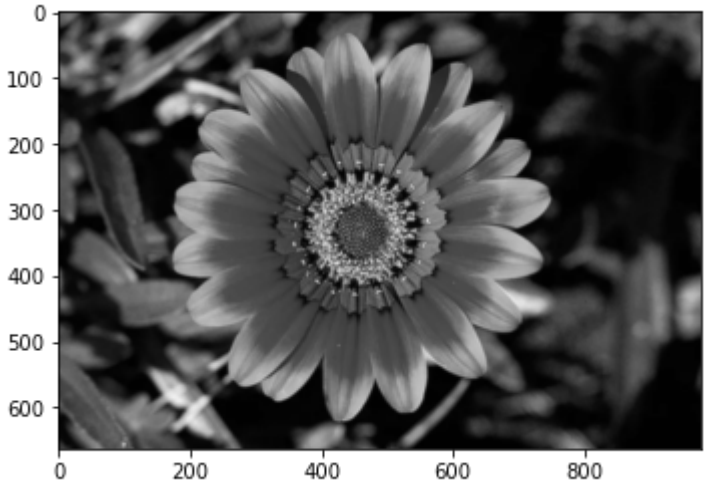


In [41]:

```
#Gaussian blurr kernal
import numpy as np
filter=np.array([[1,4,6,4,1],[4,16,24,16,4],[6,24,36,24,6],[4,16,24,16,4],[1,4,6,4,1]])/256
Gaussian_blurr_kernal=cv2.filter2D(gray_img,-1,filter)
plt.imshow(Gaussian_blurr_kernal,cmap="gray")
```

Out[41]:

```
<matplotlib.image.AxesImage at 0x2435f14c340>
```



Video Detection

In [42]:

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

In [43]:

```
cap = cv2.VideoCapture(0)

while True:
    ret,frame=cap.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_RGBA2GRAY)
    edges=cv2.Canny(gray,30,100)
    cv2.imshow("edges",edges)
    cv2.imshow("gray",gray)
    if cv2.waitKey(1)==ord("q"):
        break
cap.release()
cv2.destroyAllWindows()
```

gradient

In [103]:

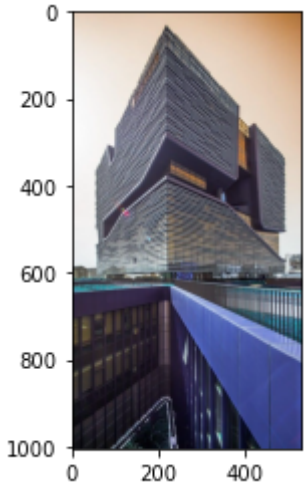
```
img=cv2.imread("8469467c1b89772eadde16bea201e901.jpg")
```

In [104]:

```
#plt.axis("off")
plt.imshow(img)
```

Out[104]:

```
<matplotlib.image.AxesImage at 0x2436237d490>
```

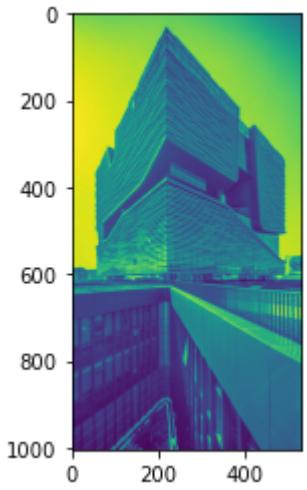


In [105]:

```
RGB_img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
plt.imshow(RGB_img)
```

Out[105]:

```
<matplotlib.image.AxesImage at 0x24362cb4160>
```

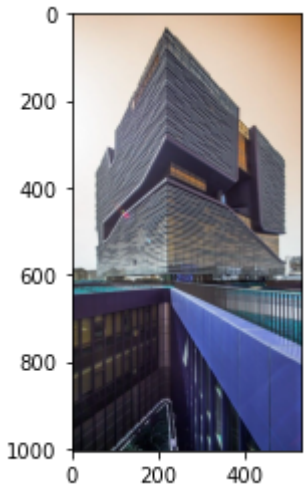


In [106]:

```
#gray_img
gray_img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
plt.imshow(img,cmap="gray")
```

Out[106]:

```
<matplotlib.image.AxesImage at 0x24361c8c7c0>
```



In [107]:

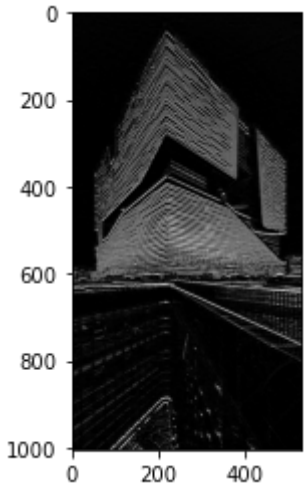
```
import numpy as np
sobel_filter_h=np.array([[-1,-2,-1],[0,0,0],[1,2,1]])
```

In [108]:

```
G_img=cv2.filter2D(gray_img,-1,sobel_filter_h)
plt.imshow(G_img,cmap="gray")
```

Out[108]:

```
<matplotlib.image.AxesImage at 0x24361f81af0>
```



In [109]:

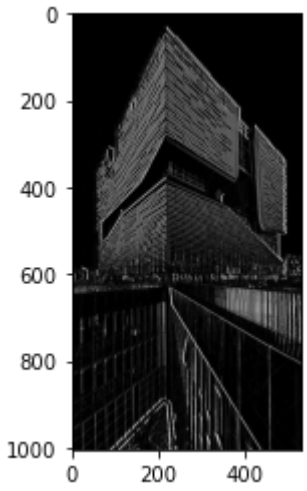
```
import numpy as np
sobel_filter_v=np.array([[-1,0,1],[-2,0,2],[-1,0,1]])
```

In [110]:

```
G_img=cv2.filter2D(gray_img,-1,sobel_filter_v)
plt.imshow(G_img,cmap="gray")
```

Out[110]:

```
<matplotlib.image.AxesImage at 0x24361b48d30>
```



In [111]:

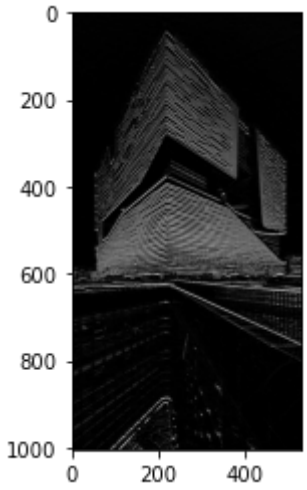
```
#
import numpy as np
sobel_filter_ph=np.array([[-1,-1,-1],[0,0,0],[1,1,1]])
```

In [112]:

```
BG_img=cv2.filter2D(gray_img,-1,sobel_filter_ph)
plt.imshow(BG_img,cmap="gray")
```

Out[112]:

```
<matplotlib.image.AxesImage at 0x24361e17f40>
```



In [ ]:

```

```

In [113]:

```
#sobel filter
import numpy as np
sobel_filter_pv=np.array([[-1,0,-1],[-1,0,1],[-1,0,-1]])
```

In [ ]:

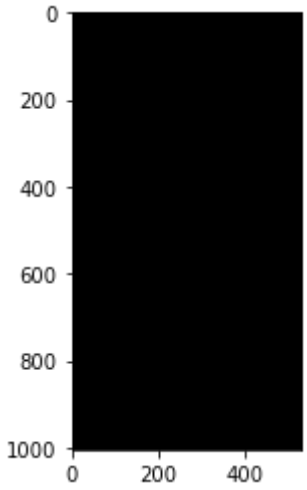
```

```

In [55]:

```
BG_img=cv2.filter2D(gray_img,-1,sobel_filter_pv)
plt.imshow(BG_img,cmap="gray")
```

Out[55]:

```
<matplotlib.image.AxesImage at 0x24363f653a0>
```



In [114]:

```
#robert
```

In [115]:

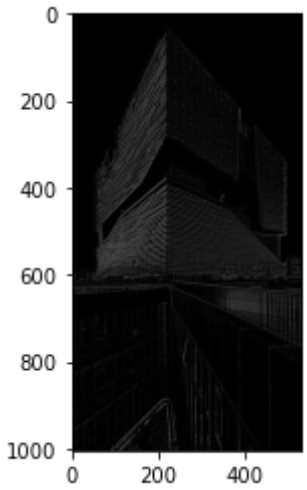
```
import numpy as np
sobel_filter_rv=np.array([[-1,0],[0,1]])
```

In [116]:

```
rG_img=cv2.filter2D(gray_img,-1,sobel_filter_rv)
plt.imshow(rG_img,cmap="gray")
```

Out[116]:

```
<matplotlib.image.AxesImage at 0x24362c913a0>
```



In [117]:

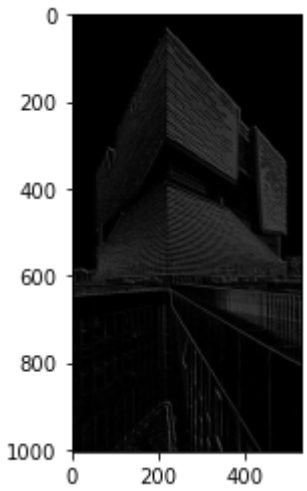
```
#Robert_filter
import numpy as np
sobel_filter_rh=np.array([[0,1],[-1,0]])
```

In [118]:

```
rh_img=cv2.filter2D(gray_img,-1,sobel_filter_rh)
plt.imshow(rh_img,cmap="gray")
```

Out[118]:

```
<matplotlib.image.AxesImage at 0x2436247e250>
```



concer dection

In [78]:

```
img=cv2.imread("th.jpg")
```

In [79]:

```
RBG_image=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.imshow(RBG_image)
```

Out[79]:

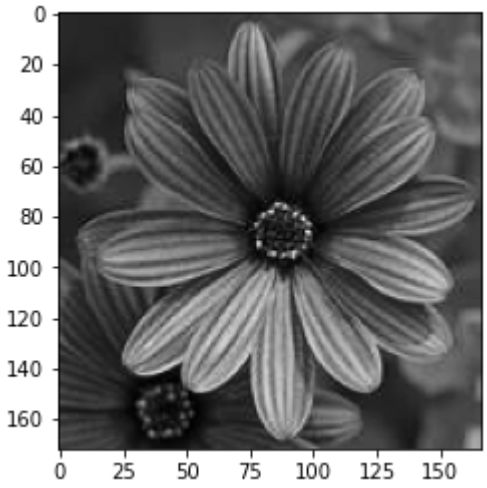
```
<matplotlib.image.AxesImage at 0x243620ad910>
```

In [80]:

```
#gray_img
gray_img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
plt.imshow(gray_img,cmap="gray")
```

Out[80]:

```
<matplotlib.image.AxesImage at 0x243627c54f0>
```



In [88]:

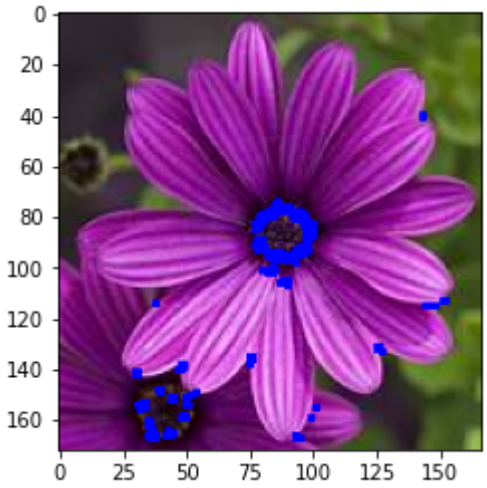
```
gray = np.float32(gray)
dst = cv2.cornerHarris(gray_img,2,3,0.04)

dst = cv2.dilate(dst,None)
```

In [89]:

```
RBG_image[dst>0.05*dst.max()]=[0,0,255]
plt.imshow(RBG_image)
```

Out[89]:

```
<matplotlib.image.AxesImage at 0x24361bf34f0>
```

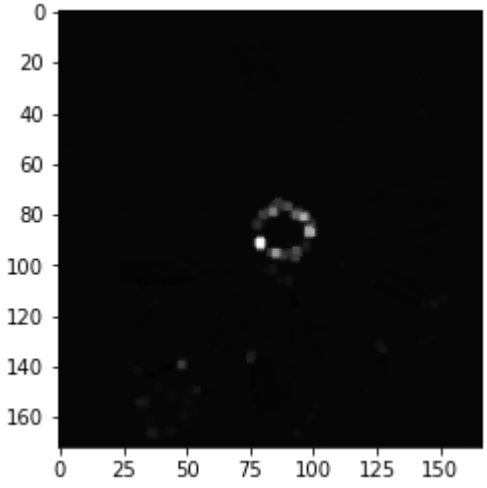


In [90]:

```
plt.imshow(dst,cmap="gray")
```

Out[90]:

```
<matplotlib.image.AxesImage at 0x24361c7fa90>
```

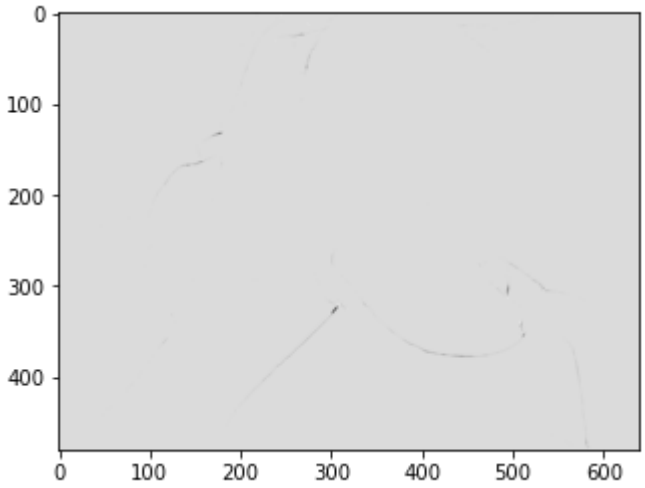


In [91]:

```
plt.imshow(dst1,cmap="gray")
```

Out[91]:

```
<matplotlib.image.AxesImage at 0x24361b6f160>
```



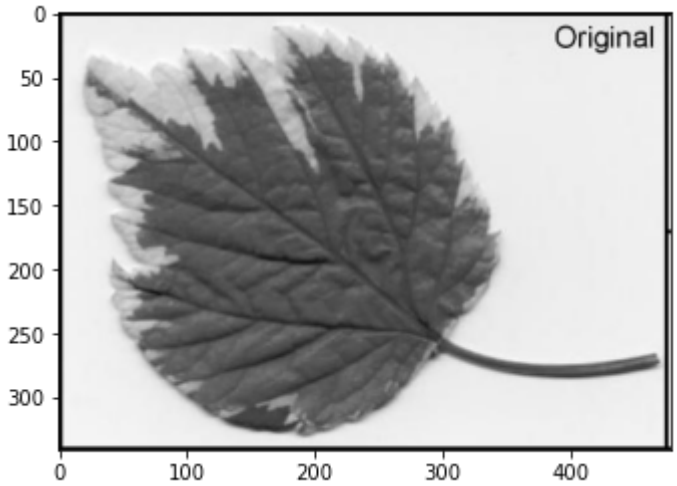
Morphology

In [96]:

```
img=cv2.imread("gray-morphology.jpg")
plt.imshow(img)
```

Out[96]:

```
<matplotlib.image.AxesImage at 0x2436242b3a0>
```



In [97]:

```
#dilate image
kernel = np.ones((5, 5), 'uint8')

dilate_img = cv2.dilate(img, kernel, iterations=2)
```

In [98]:

```
#erosion image


erode_img = cv2.erode(img, kernel, iterations=2)
```

In [99]:

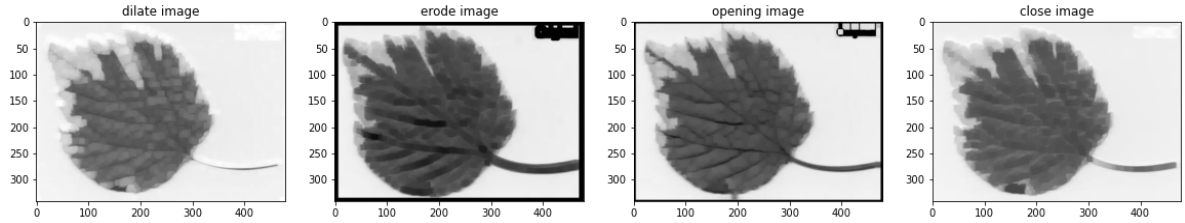
```
#opening
opening=cv2.morphologyEx(img,cv2.MORPH_OPEN,kernel, iterations=2)
```

In [100]:

```
#closing
closing=cv2.morphologyEx(img,cv2.MORPH_CLOSE,kernel, iterations=2)
```

In [101]:

```
plt.figure(figsize=[20,5])
plt.subplot(141);plt.imshow(dilate_img);plt.title("dilate image");

plt.subplot(142);plt.imshow(erode_img);plt.title("erode image");

plt.subplot(143);plt.imshow(opening);plt.title("opening image");

plt.subplot(144);plt.imshow(closing);plt.title("close image");
```



In [102]:

```
import cv2
import sys
import numpy
```

In [ ]:

```
PREVIEW=0
BLUR=1
FEATURES=2
CANNY=3
```

In [ ]:

```
feature_params=dict(maxCorners=500,
                    qualitylevel=0.2,
                    minDISTANCE=15,
                    blockSIZE=9)
```

In [ ]:

```
s=0
if len(sys.argv)>1:
    s=sys.argv[1]

image_fliter=PREVIEW
alive=True
win_name="camera fliter"
cv2.namedWindow(win_name,cv2.WINDOW_NORMAL)
result=None
source=cv2.VideoCapture(s)
while alive:
    has_frame=source.read()
    if not has_frame:
        break


    frame=cv2.flip(frame,1)

    if image_fliter==PREVIEW:
        result=frame
    elif image_fliter==CANNY:
         result=cv2.Canny(frame,145,150)
    elif image_fliter==BLUR:
         result=cv2.blur(frame,(13,13))
    elif image_fliter==FEATURES:
         result=frame
         frame_gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
         corners=cv2.goodFeaturesToTrack(frame_gray,**feature_params)
        if corners is not None:
            for x,y in numpy.float32(corners).reshape(-1,2):
            cv2.circle(result,(x,y),10,(0,255,0),1)
    cv2.imshow(win_name,result)
    key=cv2.waitKey(1)
    if key==ord("Q")or key == ord("q") or key==27:
        alive=False
    elif key==ord("C")or key ==ord("c"):
        image_fliter=CANNY
    elif key==ord("B")or key ==ord("b"):
        image_fliter=BLUR
    elif key==ord("F")or key ==ord("f"):
        image_fliter=FEATURES
    elif key==ord("P")or key ==ord("p"):
        image_fliter=PREVIEW


source.release()
cv2.destroyWindow(win_name)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: