EXPLORING GEN AI THROUGH COMPUTER VISION

Human Vision (vs) Computer Vision:

Human Vision: Human sees the object through eyes, sends signals to brain and identifys the object name and produces the desired output . Computer Vision Computer identifies the objects/images through sensing devices using interpreting devices and produces the outpuhite) lue t.

Actual Image – it is represented in the form of array. Then, converted into the form of pixels as 0's and 1's. Pixel ranges between 0 to 255. 0 represents darkest value. 255 represents highest or brightest value.

The coloured images are Red, Green, Blue. Also can use combination of colours to get different colours. 2D channel -- black and white. 3D channel – Red, Green, Blue. Grey_Scale images are – black and white. Coloured images represents -- Colourful images (exluding black and white).

In [ ]:
```
import numpy as np #numpy is library for nd array
```

In [2]:
```
ones_arr = np.ones((5,5)) #image is broken into array format - rows and columns
```

In [3]:
```
ones_arr
```

Out[3]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [4]:
```
#Ignoring float values and need int values
ones_arr = np.ones((5,5),dtype = int)
```

In [5]:
```
ones_arr
```

Out[5]:
```
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]])
```

In [6]:
```
zeros_arr = np.zeros((3,3),dtype = int)
```

In [7]:
```
zeros_arr
```

Out[7]:
```
array([[0, 0, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

In [8]:
```
ones_arr
```

Out[8]:
```
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]])
```

In [9]:
```
#entire 1's array converted to 255
ones_arr * 255
```

Out[9]:
```
array([[255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255],
       [255, 255, 255, 255, 255]])
```

In [10]:
```
ones_arr
```

Out[10]:
```
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]])
```

In [11]:
```
import matplotlib.pyplot as plt #matplotlib is library for visualization
```

In [12]:
```
# all the pictures are kept inside the graph or box
%matplotlib inline
```

In [13]:
```
# PIL represents - Python Imaging Library
# When we work with images we need to use PIL
from PIL import Image
```

In [14]:
```
#We have downloaded the image from google saved in the below location and now readi
flower_img = Image.open(r'C:\Users\SUNITHA\Desktop\flower.jpg')
```

In [15]:
```
flower_img
```

Out[15]:

In [16]:
```
# Finding the type of image we are using
type(flower_img)
#So the image shows the output as PIL.JpegImagePlugin datatype
```

Out[16]:
```
PIL.JpegImagePlugin.JpegImageFile
```

In [17]:
```
#Finding the array format of this image
flower_arr = np.asarray(flower_img)
flower_arr
#Below the entire image is converted in the form of array
```

Out[17]:
```
array([[[234, 235, 229],
        [232, 233, 227],
        [229, 230, 224],
        ...,
        [196, 174, 160],
        [198, 175, 161],
        [201, 175, 160]],

       [[233, 234, 228],
        [230, 231, 225],
        [229, 228, 223],
        ...,
        [192, 169, 153],
        [194, 168, 151],
        [195, 168, 151]],

       [[232, 233, 227],
        [229, 230, 224],
        [227, 226, 221],
        ...,
        [188, 161, 140],
        [188, 160, 138],
        [189, 158, 137]],

       ...,

       [[180, 152, 149],
        [177, 149, 146],
        [174, 146, 143],
        ...,
        [175, 165, 173],
        [178, 171, 178],
        [180, 173, 180]],

       [[184, 154, 154],
        [180, 152, 151],
        [176, 148, 145],
        ...,
        [174, 164, 172],
        [178, 168, 176],
        [180, 170, 178]],

       [[185, 155, 155],
        [180, 152, 151],
        [176, 148, 147],
        ...,
        [173, 164, 169],
        [176, 167, 172],
        [178, 169, 174]]], dtype=uint8)
```

In [18]:
```
#Finding the type of flower_arr
type(flower_arr)
#As the type of flower_arr is in the form of arrays it displays the type as numpy.n
```

Out[18]:
```
numpy.ndarray
```

shape represents dimensions, in below output

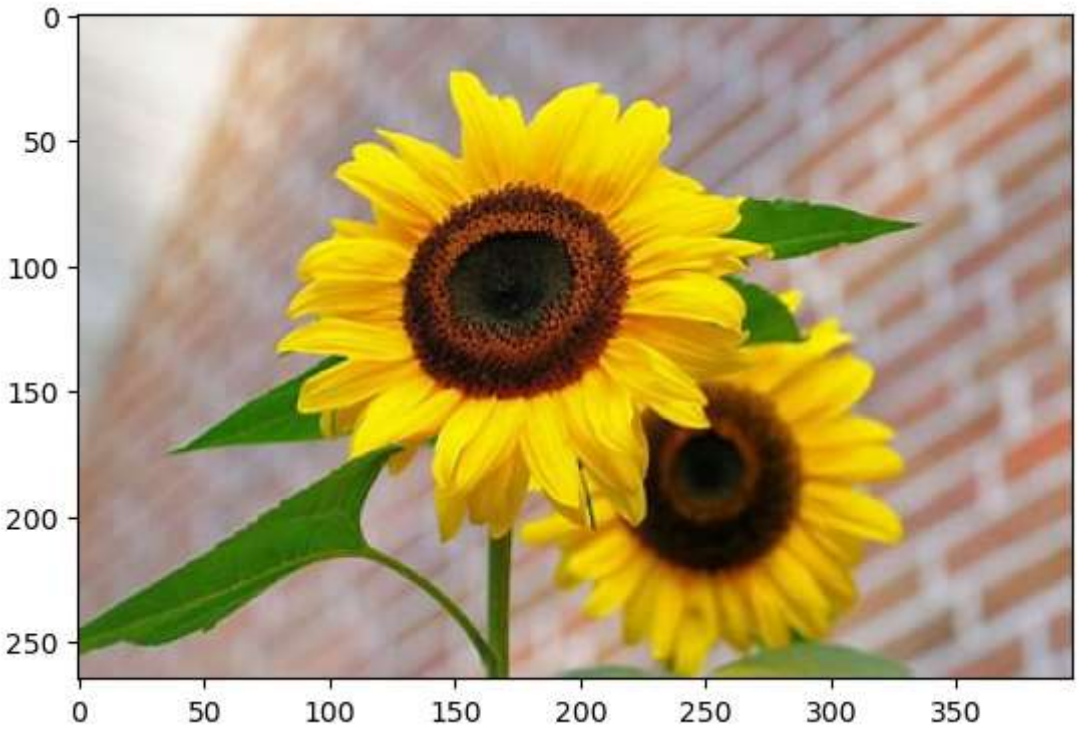
- 266 - represents width
- 474 - represents height
- 3 - represents 3D channel (3D channel - means Red, Green, Blue)

In [20]:
```
flower_arr.shape
```

Out[20]:
```
(265, 397, 3)
```

In [21]:
```
plt.imshow(flower_arr)
```

Out[21]:
```
<matplotlib.image.AxesImage at 0x1b3058e2bd0>
```



In [22]:
```
#Lets make changes to the above image so we shall copy it first
flower_red = flower_arr.copy()
```

In [23]:
```
flower_red
```

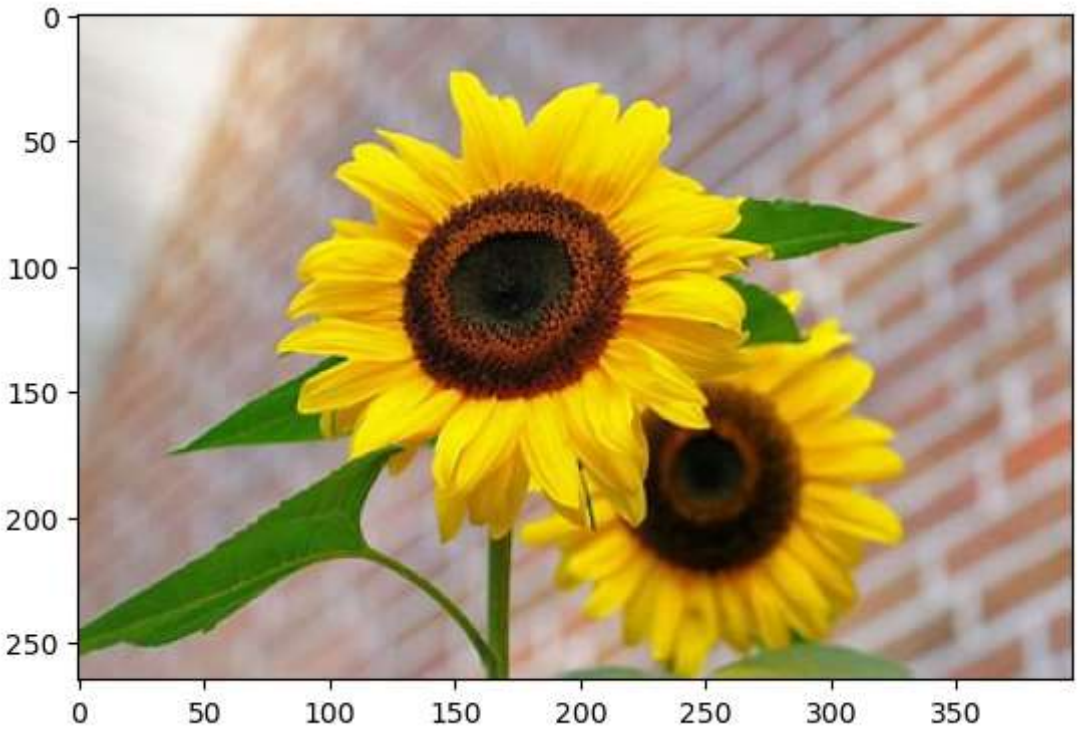
Out[23]:
```
array([[[234, 235, 229],
        [232, 233, 227],
        [229, 230, 224],
        ...,
        [196, 174, 160],
        [198, 175, 161],
        [201, 175, 160]],

       [[233, 234, 228],
        [230, 231, 225],
        [229, 228, 223],
        ...,
        [192, 169, 153],
        [194, 168, 151],
        [195, 168, 151]],

       [[232, 233, 227],
        [229, 230, 224],
        [227, 226, 221],
        ...,
        [188, 161, 140],
        [188, 160, 138],
        [189, 158, 137]],

       ...,

       [[180, 152, 149],
        [177, 149, 146],
        [174, 146, 143],
        ...,
        [175, 165, 173],
        [178, 171, 178],
        [180, 173, 180]],

       [[184, 154, 154],
        [180, 152, 151],
        [176, 148, 145],
        ...,
        [174, 164, 172],
        [178, 168, 176],
        [180, 170, 178]],

       [[185, 155, 155],
        [180, 152, 151],
        [176, 148, 147],
        ...,
        [173, 164, 169],
        [176, 167, 172],
        [178, 169, 174]]], dtype=uint8)
```

In [24]:
```
#Checking the conditions if array image is equal to the array copy image
#use double equal (==)
flower_arr == flower_red
#Here all are True means the array image and copy image are the same
```

Out[24]:
```
array([[[ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True],
        ...,
        [ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True]],

       [[ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True],
        ...,
        [ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True]],

       [[ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True],
        ...,
        [ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True]],

       ...,

       [[ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True],
        ...,
        [ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True]],

       [[ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True],
        ...,
        [ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True]],

       [[ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True],
        ...,
        [ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True]]])
```

In [25]:
```
plt.imshow(flower_red)
```

Out[25]:
```
<matplotlib.image.AxesImage at 0x1b3059f1f40>
```

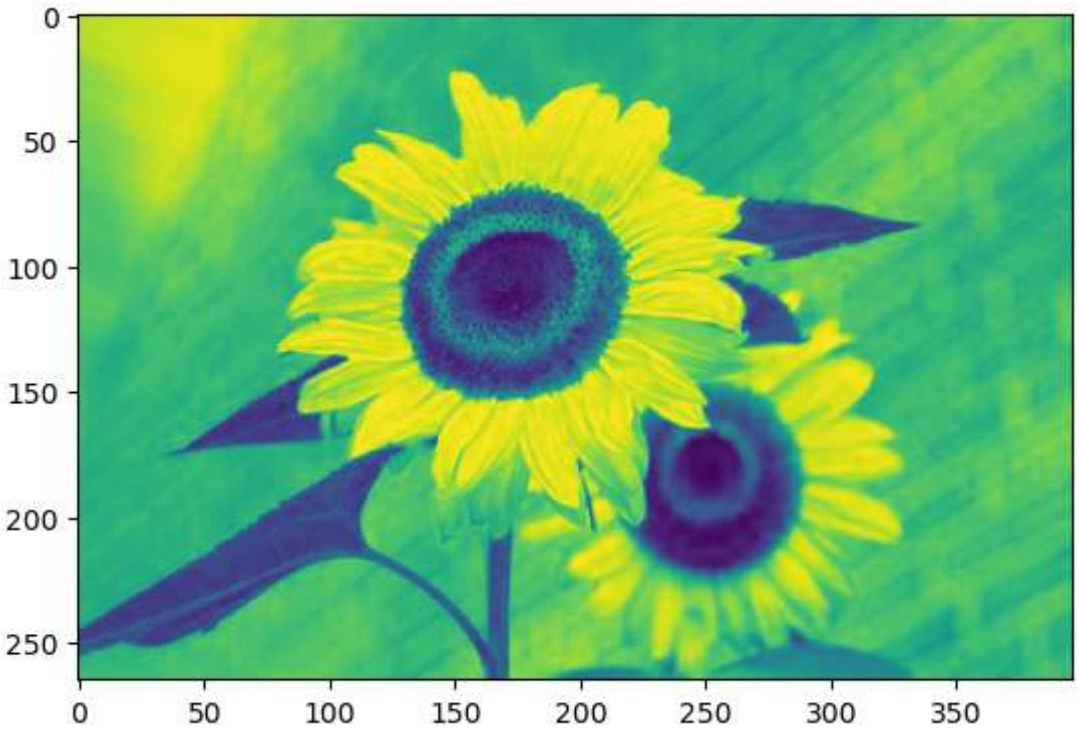


In [26]:
```
#The actual image and the above image shows a slight difference, so to make both lo
flower_red.shape
```

Out[26]:
```
(265, 397, 3)
```

In [27]:
```
#Making all rows and columns to zero of flower_red image, the image color changes
#R G B - Red,Green,Blue
plt.imshow(flower_red[:,:,0])
```

Out[27]:
```
<matplotlib.image.AxesImage at 0x1b305a69280>
```

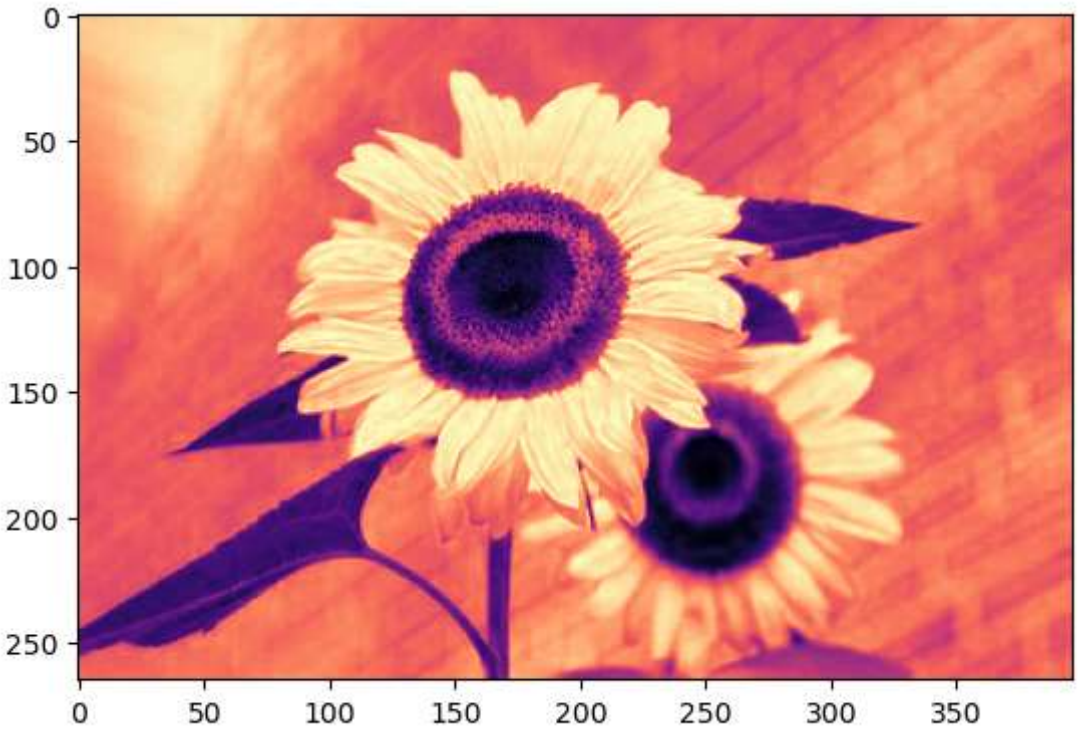


In [28]:
```
flower_red[:,:,0]
```

Out[28]:
```
array([[234, 232, 229, ..., 196, 198, 201],
       [233, 230, 229, ..., 192, 194, 195],
       [232, 229, 227, ..., 188, 188, 189],
       ...,
       [180, 177, 174, ..., 175, 178, 180],
       [184, 180, 176, ..., 174, 178, 180],
       [185, 180, 176, ..., 173, 176, 178]], dtype=uint8)
```

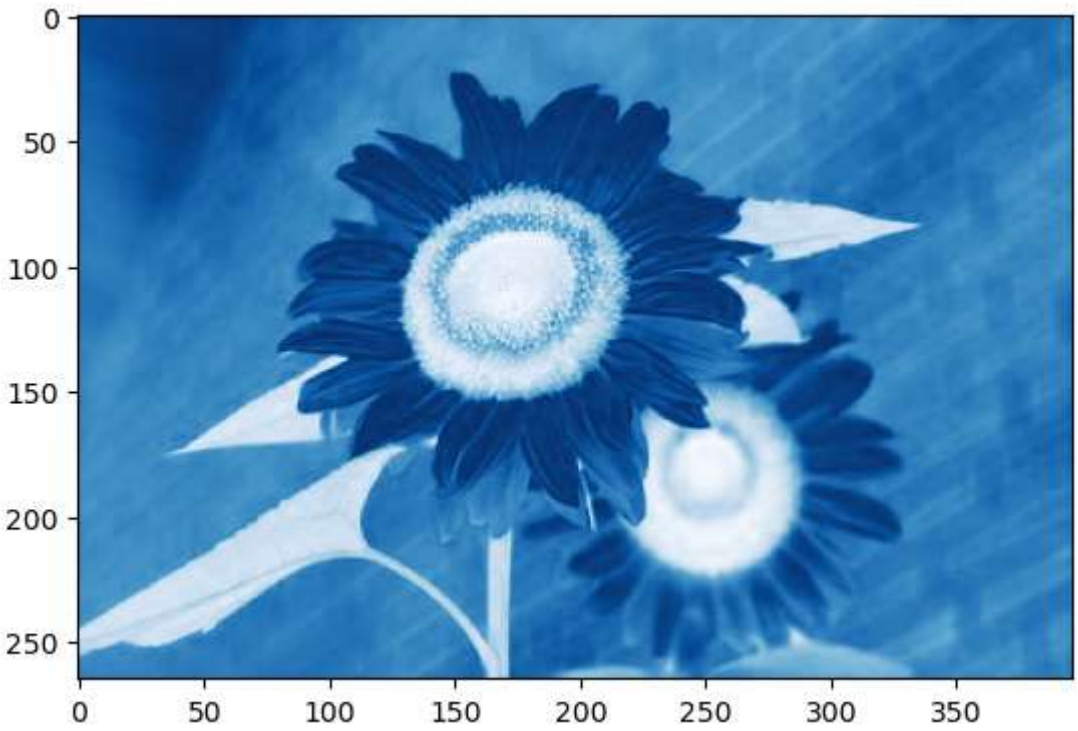
In [29]:
```
plt.imshow(flower_red[:,:,0],cmap = 'magma')
```

Out[29]:
```
<matplotlib.image.AxesImage at 0x1b305a68fe0>
```



In [30]:
```
plt.imshow(flower_red[:,:,0],cmap = 'Blues')
```

Out[30]: <matplotlib.image.AxesImage at 0x1b305b35a30>



In [31]:
```
plt.imshow(flower_red[:,:,0],cmap = 'YlGn')
```

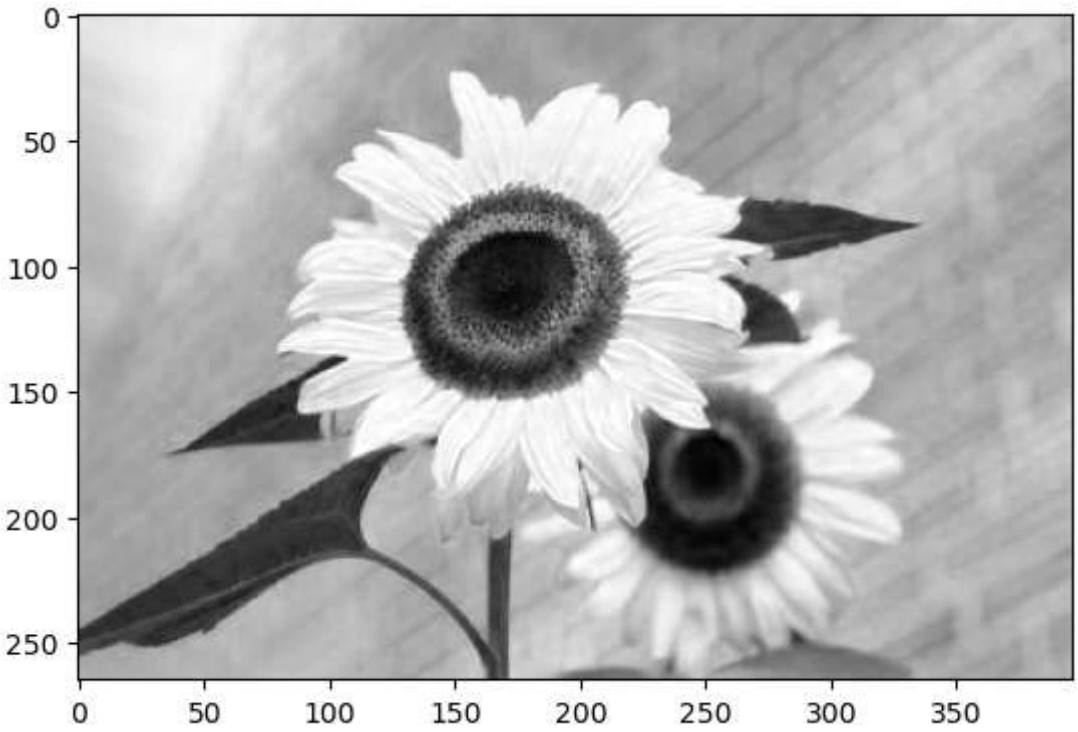
Out[31]: <matplotlib.image.AxesImage at 0x1b3073aec00>

In [32]:
```
plt.imshow(flower_red[:,:,0],cmap = 'Greys')
```

Out[32]: <matplotlib.image.AxesImage at 0x1b307422d50>

In [33]:
```
plt.imshow(flower_red[:,:,0],cmap = 'grey') #here we are using value 0
```

Out[33]: <matplotlib.image.AxesImage at 0x1b305aebe00>

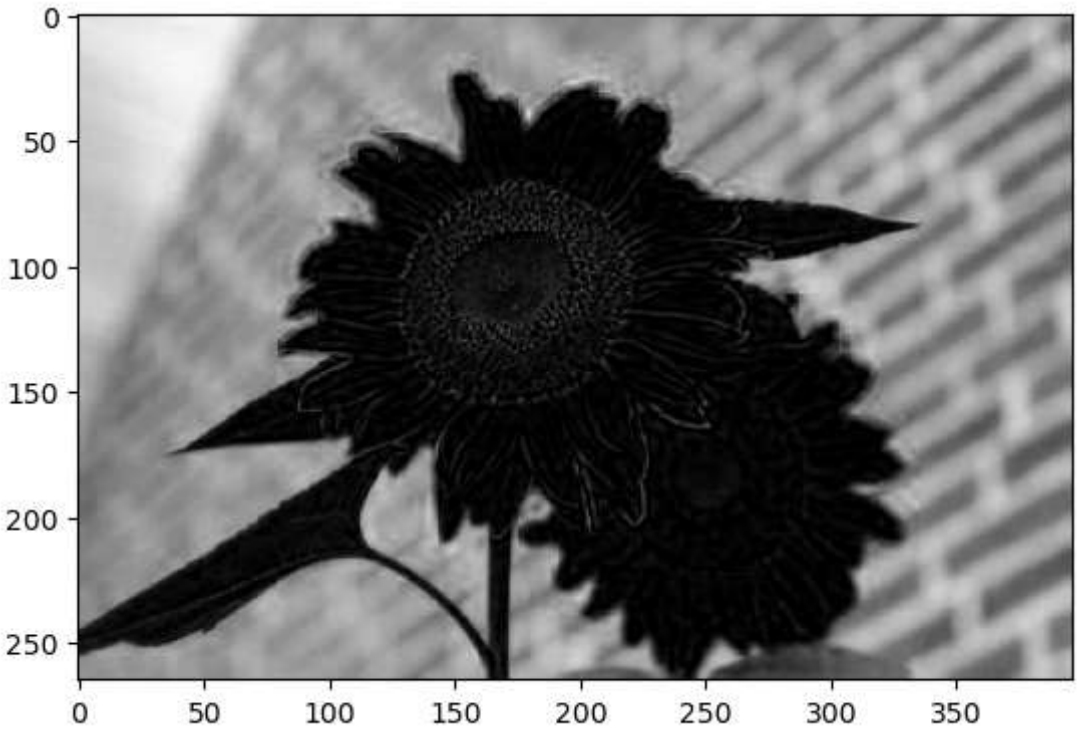


In [34]:
```
plt.imshow(flower_red[:,:,1],cmap = 'grey') #here we are using value 1
```

Out[34]: <matplotlib.image.AxesImage at 0x1b30748b440>

In [35]:
```
plt.imshow(flower_red[:,:,2],cmap = 'grey') #here we are using value 2
```

Out[35]: <matplotlib.image.AxesImage at 0x1b30756a960>



In [36]:
```
flower_red[:,:,0] #values change in the output while using 0 or 1 or 2
```

Out[36]:
```
array([[234, 232, 229, ..., 196, 198, 201],
       [233, 230, 229, ..., 192, 194, 195],
       [232, 229, 227, ..., 188, 188, 189],
       ...,
       [180, 177, 174, ..., 175, 178, 180],
       [184, 180, 176, ..., 174, 178, 180],
       [185, 180, 176, ..., 173, 176, 178]], dtype=uint8)
```

In [37]:
```
flower_red[:,:,1]
```

Out[37]:
```
array([[235, 233, 230, ..., 174, 175, 175],
       [234, 231, 228, ..., 169, 168, 168],
       [233, 230, 226, ..., 161, 160, 158],
       ...,
       [152, 149, 146, ..., 165, 171, 173],
       [154, 152, 148, ..., 164, 168, 170],
       [155, 152, 148, ..., 164, 167, 169]], dtype=uint8)
```

In [38]:
```
flower_red[:,:,2]
```

Out[38]:
```
array([[229, 227, 224, ..., 160, 161, 160],
       [228, 225, 223, ..., 153, 151, 151],
       [227, 224, 221, ..., 140, 138, 137],
       ...,
       [149, 146, 143, ..., 173, 178, 180],
       [154, 151, 145, ..., 172, 176, 178],
       [155, 151, 147, ..., 169, 172, 174]], dtype=uint8)
```

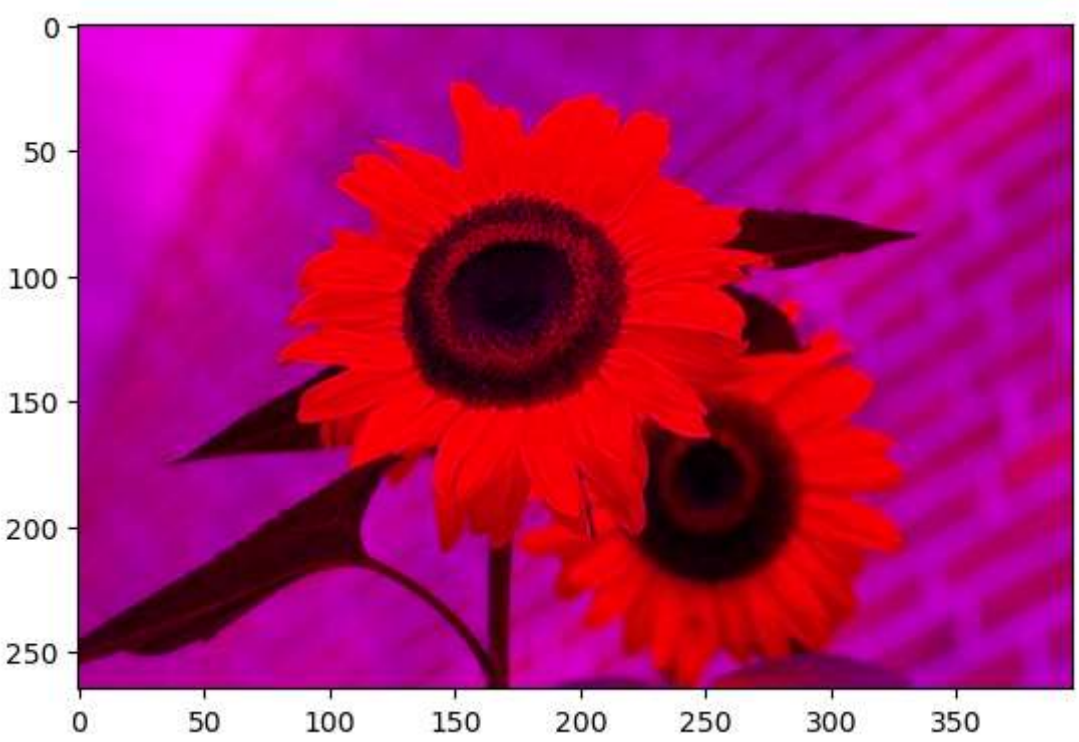
In [39]:
```
flower_red[:,:,1] = 0  #Changing image arrays values to zero
```

In [40]:
```
flower_red[:,:,1]
```

Out[40]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=uint8)
```

In [41]:
```
plt.imshow(flower_red)
```

Out[41]:
```
<matplotlib.image.AxesImage at 0x1b3062ccc20>
```



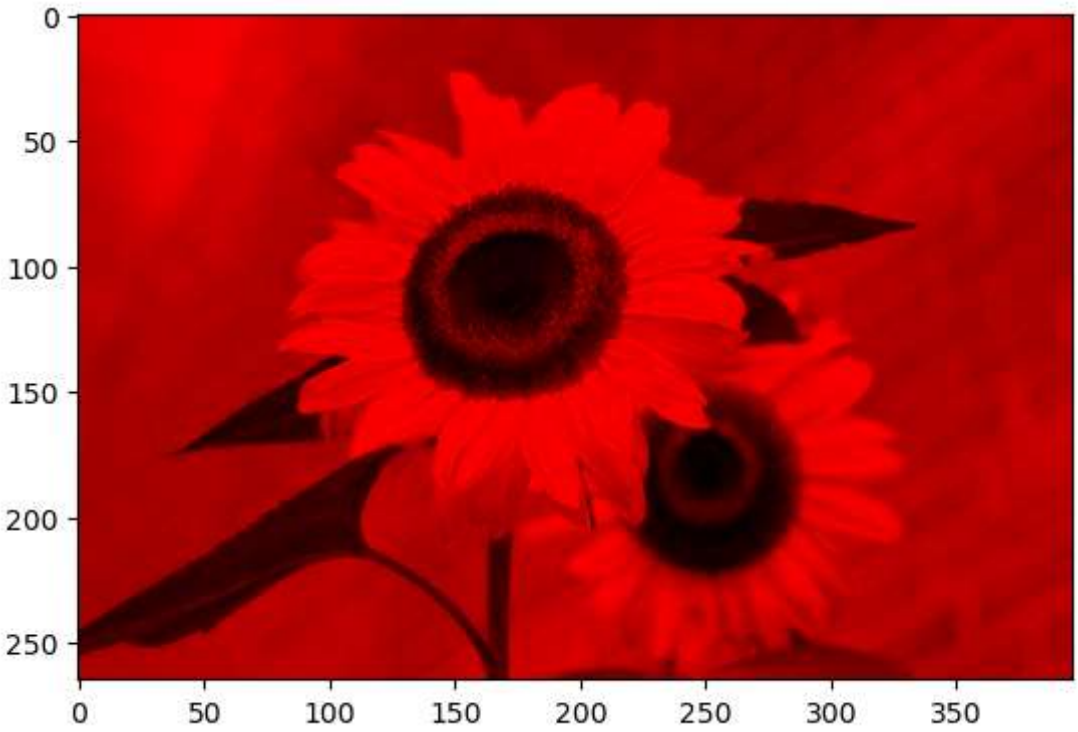
In [42]:
```
flower_red[:,:,2] = 0  #Changing image arrays values to zero
```

In [43]:
```
flower_red[:,:,2]
```

Out[43]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=uint8)
```

In [44]:
```
plt.imshow(flower_red)
```

Out[44]:
```
<matplotlib.image.AxesImage at 0x1b30752a8a0>
```



In [45]:
```
flower_arr
```

Out[45]:
```
array([[[234, 235, 229],
        [232, 233, 227],
        [229, 230, 224],
        ...,
        [196, 174, 160],
        [198, 175, 161],
        [201, 175, 160]],

       [[233, 234, 228],
        [230, 231, 225],
        [229, 228, 223],
        ...,
        [192, 169, 153],
        [194, 168, 151],
        [195, 168, 151]],

       [[232, 233, 227],
        [229, 230, 224],
        [227, 226, 221],
        ...,
        [188, 161, 140],
        [188, 160, 138],
        [189, 158, 137]],

       ...,

       [[180, 152, 149],
        [177, 149, 146],
        [174, 146, 143],
        ...,
        [175, 165, 173],
        [178, 171, 178],
        [180, 173, 180]],

       [[184, 154, 154],
        [180, 152, 151],
        [176, 148, 145],
        ...,
        [174, 164, 172],
        [178, 168, 176],
        [180, 170, 178]],

       [[185, 155, 155],
        [180, 152, 151],
        [176, 148, 147],
        ...,
        [173, 164, 169],
        [176, 167, 172],
        [178, 169, 174]]], dtype=uint8)
```

In [46]:
```
flower_red
```

Out[46]:
```
array([[[234,   0,   0],
        [232,   0,   0],
        [229,   0,   0],
        ...,
        [196,   0,   0],
        [198,   0,   0],
        [201,   0,   0]],

       [[233,   0,   0],
        [230,   0,   0],
        [229,   0,   0],
        ...,
        [192,   0,   0],
        [194,   0,   0],
        [195,   0,   0]],

       [[232,   0,   0],
        [229,   0,   0],
        [227,   0,   0],
        ...,
        [188,   0,   0],
        [188,   0,   0],
        [189,   0,   0]],

       ...,

       [[180,   0,   0],
        [177,   0,   0],
        [174,   0,   0],
        ...,
        [175,   0,   0],
        [178,   0,   0],
        [180,   0,   0]],

       [[184,   0,   0],
        [180,   0,   0],
        [176,   0,   0],
        ...,
        [174,   0,   0],
        [178,   0,   0],
        [180,   0,   0]],

       [[185,   0,   0],
        [180,   0,   0],
        [176,   0,   0],
        ...,
        [173,   0,   0],
        [176,   0,   0],
        [178,   0,   0]]], dtype=uint8)
```

In [47]:
```
flower_img
```

Out[47]:

In [48]:
```
arr1 = np.asarray(flower_img)
```

In [49]:
```
type(arr1)
```

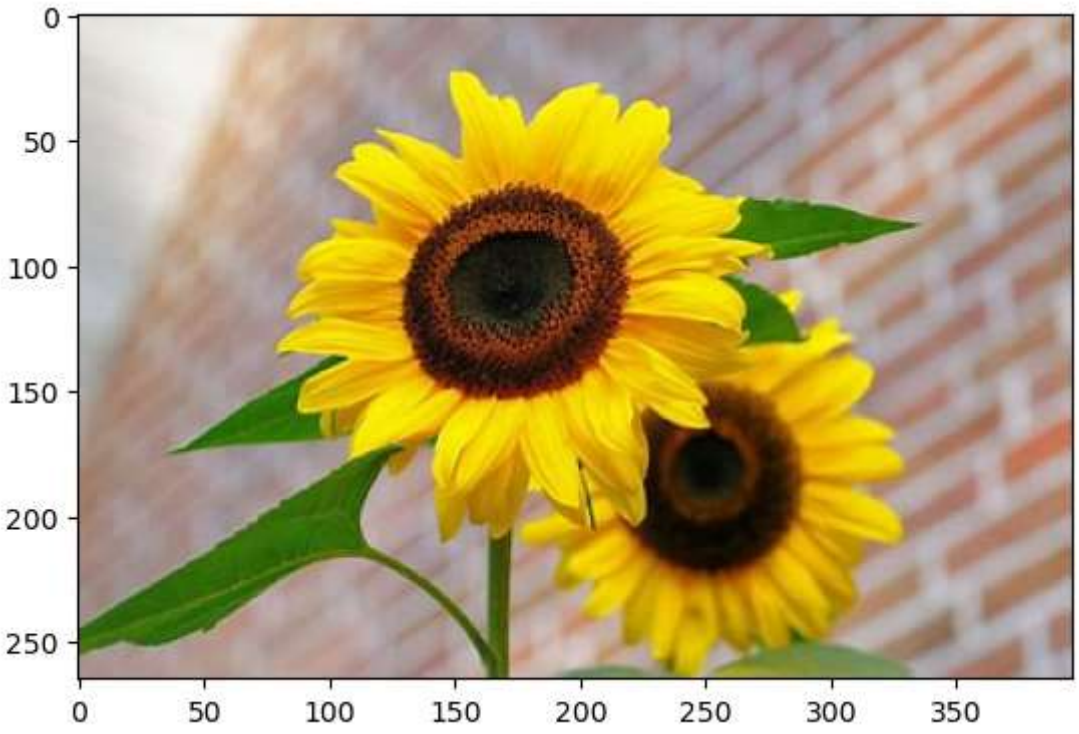
Out[49]:
```
numpy.ndarray
```

In [50]:
```
arr1.shape
```

Out[50]:
```
(265, 397, 3)
```

In [51]:
```
plt.imshow(arr1)
```

Out[51]:
```
<matplotlib.image.AxesImage at 0x1b3075d86b0>
```



In [52]:
```
flower_img1 = arr1.copy()
```

In [53]:
```
flower_img1[:,:,0] = 0
```

In [54]:
```
plt.imshow(flower_img1)
```

Out[54]:
```
<matplotlib.image.AxesImage at 0x1b30762f6e0>
```

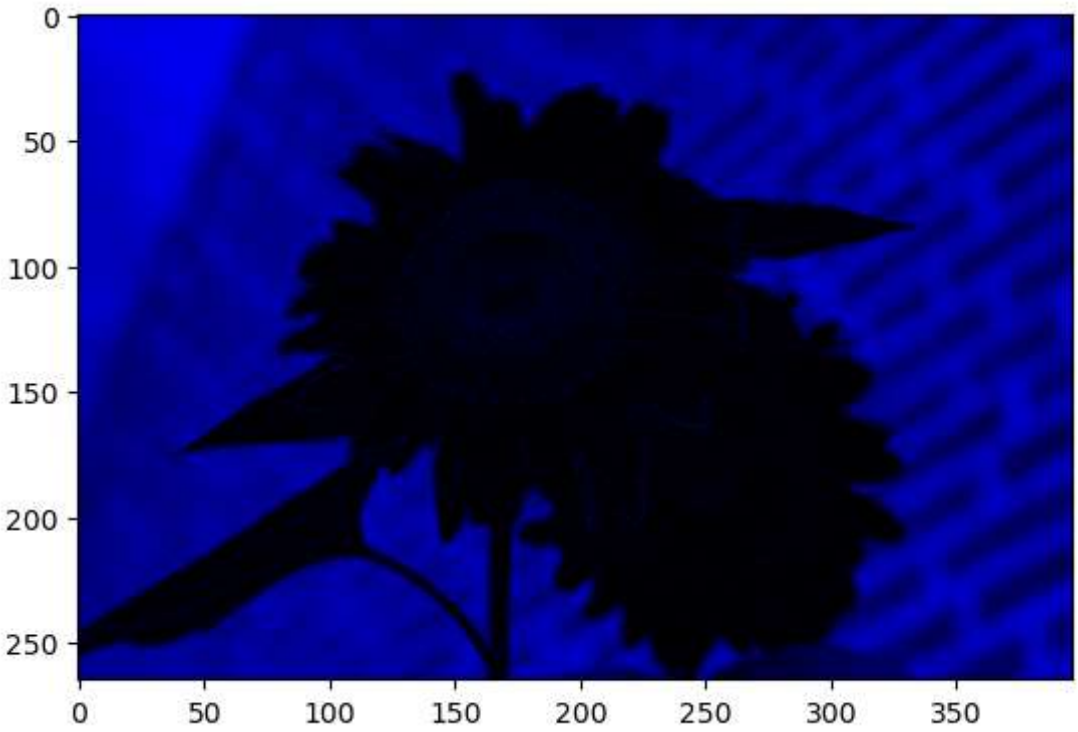
In [55]:
```
flower_img1[:,:,1]
```

Out[55]:
```
array([[235, 233, 230, ..., 174, 175, 175],
       [234, 231, 228, ..., 169, 168, 168],
       [233, 230, 226, ..., 161, 160, 158],
       ...,
       [152, 149, 146, ..., 165, 171, 173],
       [154, 152, 148, ..., 164, 168, 170],
       [155, 152, 148, ..., 164, 167, 169]], dtype=uint8)
```

In [56]:
```
flower_img1[:,:,1] = 0
```

In [57]:
```
plt.imshow(flower_img1)
```

Out[57]:
```
<matplotlib.image.AxesImage at 0x1b3077e4f50>
```



In [ ]: