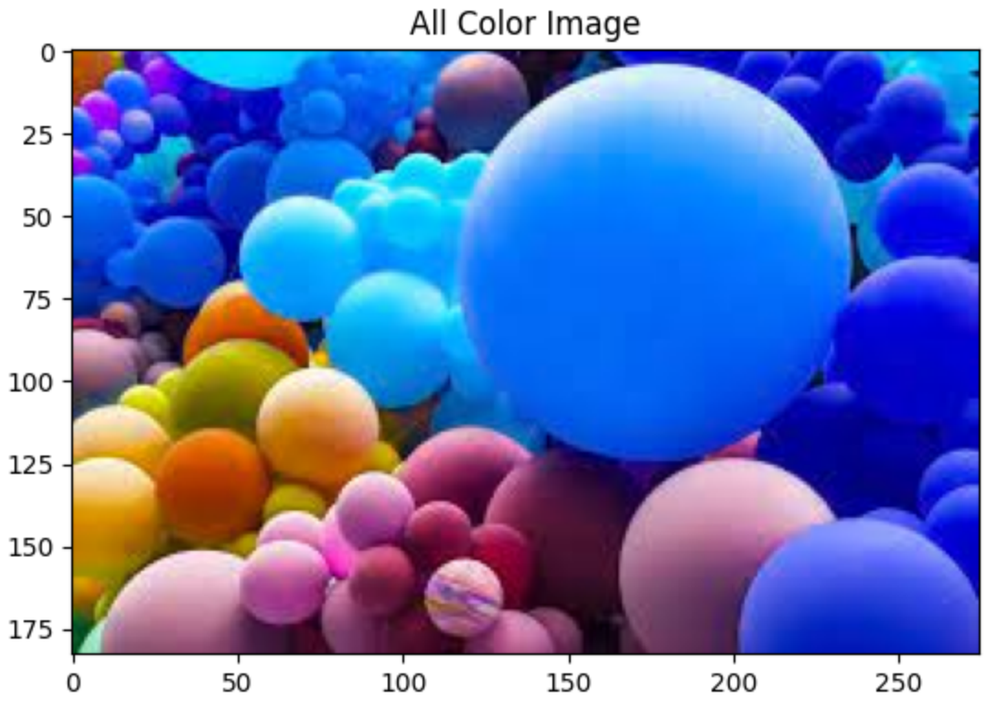
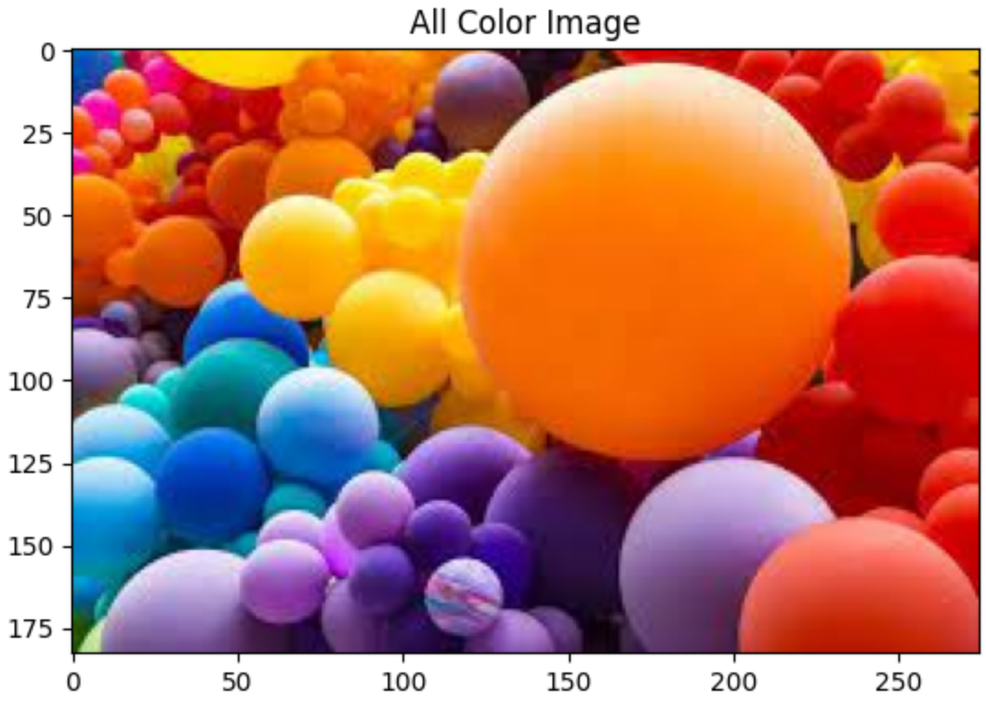
In [17]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

In [26]:
```
img1 = cv2.imread('/content/drive/MyDrive/Image Processing Humber/AllColorsIma
img2 = cv2.imread('/content/drive/MyDrive/Image Processing Humber/SunFlower.pn
img3 = cv2.imread('/content/drive/MyDrive/Image Processing Humber/boatcolor.ti
```

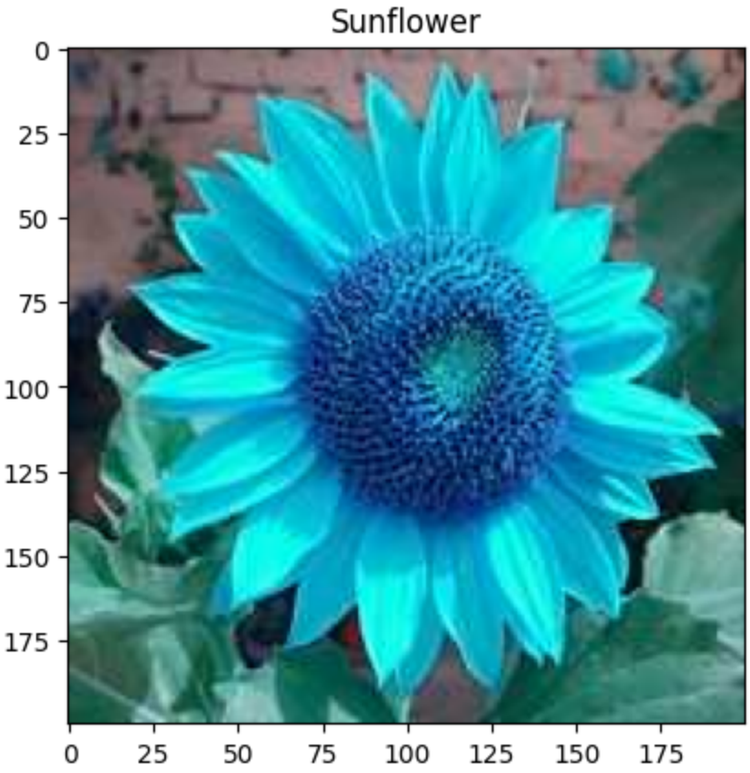
In [27]:
```
plt.imshow(img1)      # R,G,B format
plt.title('All Color Image')
plt.show()
```



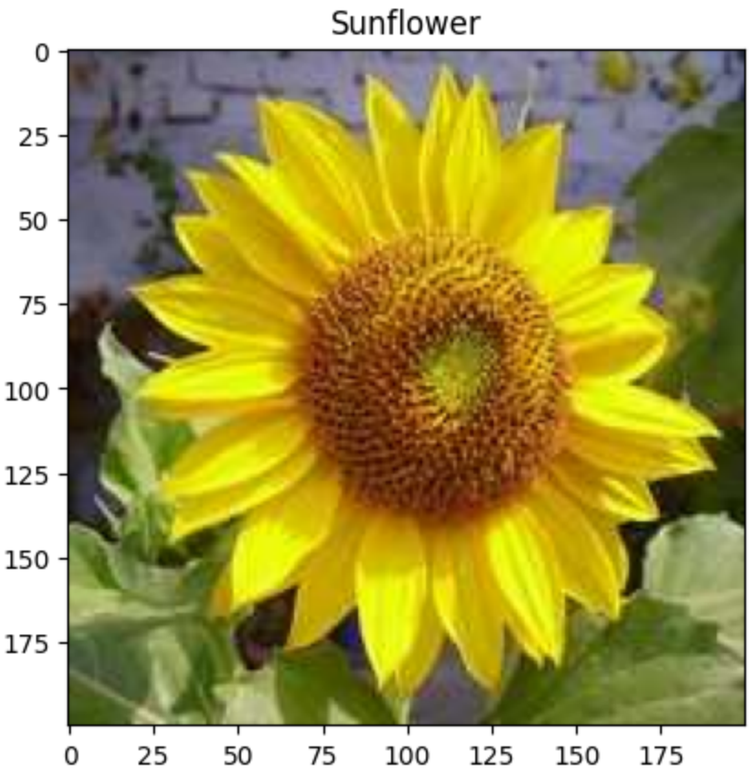
In [50]:
```
img1_RGB = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB) # to display image using matp
                                                 # BGR To RGB
plt.imshow(img1_RGB)
plt.title('All Color Image')
plt.show()
```



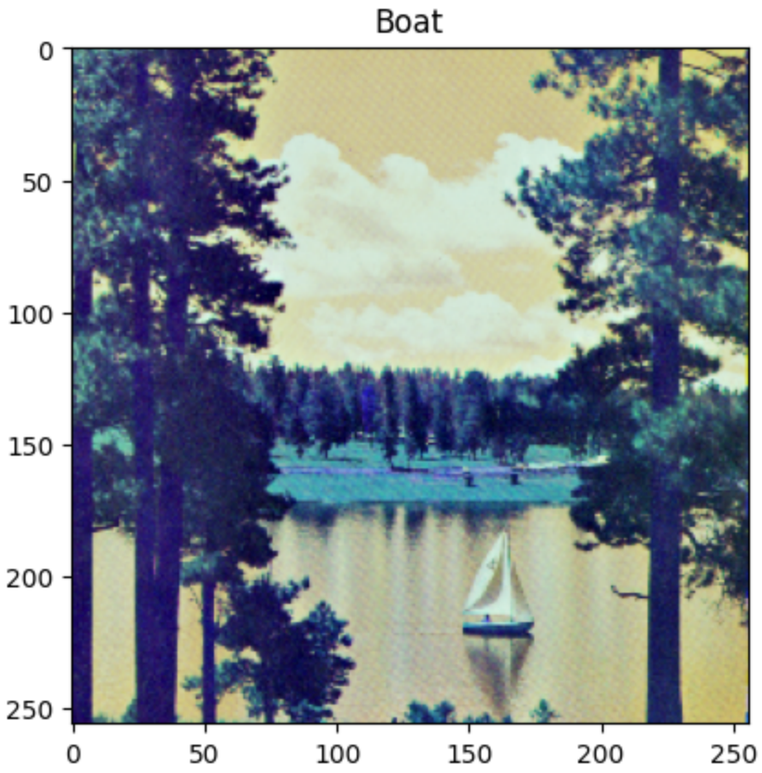
In [29]:
```
plt.imshow(img2)      # R,G,B format
plt.title('Sunflower')
plt.show()
```



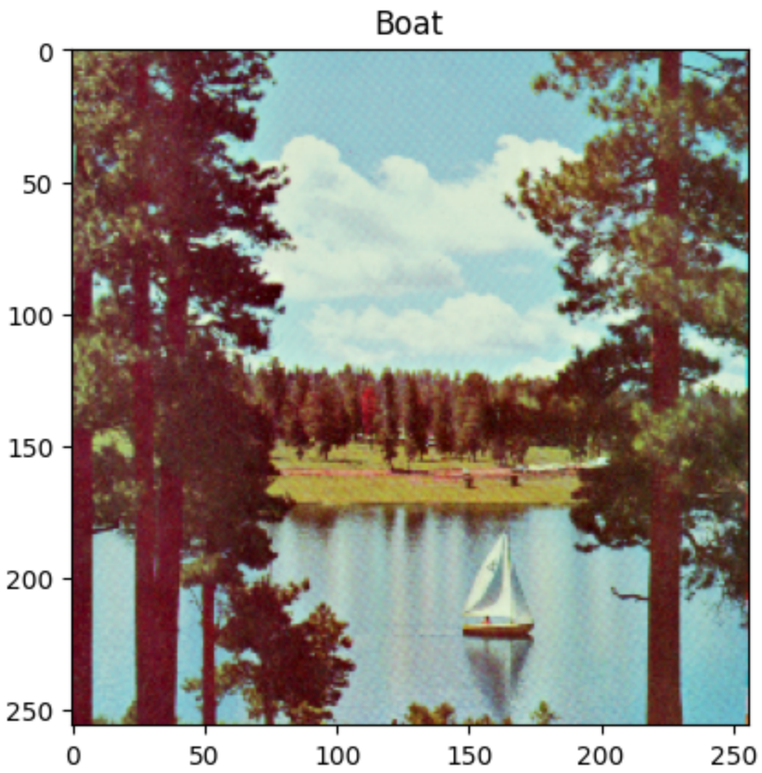
In [51]:
```
img2_RGB = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB) # to display image using matp
                                                 # BGR To RGB
plt.imshow(img2_RGB)
plt.title('Sunflower')
plt.show()
```



In [31]:
```
plt.imshow(img3)       # R,G,B format
plt.title('Boat')
plt.show()
```



In [52]:
```
img3_RGB = cv2.cvtColor(img3, cv2.COLOR_BGR2RGB) # to display image using matp
                                                 # BGR To RGB
plt.imshow(img3_RGB)
plt.title('Boat')
plt.show()
```



In [35]:
```
# Dispaly the datatype for each image

print(img1_RGB.dtype)
print(img2_RGB.dtype)
print(img3_RGB.dtype)
```

```
uint8
uint8
uint8
```

In [36]:
```
# Dispaly the dimension for each image 1

print(img1_RGB.shape)
print(" ")
print(img1_RGB.ndim) # Number of dimension. The thired value shows the channel
print(" ")
print(img1_RGB.size) # 225 x 225 x 3 = 151875
```

```
(183, 275, 3)

3

150975
```

In [37]:
```
# Dispaly the dimension for each image 2

print(img2_RGB.shape)
print(" ")
print(img2_RGB.ndim) # Number of dimension. The thired value shows the channel
print(" ")
print(img2_RGB.size) # 225 x 225 x 3 = 151875
```

```
(200, 200, 3)

3

120000
```

In [38]:
```
# Dispaly the dimension for each image 3

print(img3_RGB.shape)
print(" ")
print(img3_RGB.ndim) # Number of dimension. The thired value shows the channel
print(" ")
print(img3_RGB.size) # 225 x 225 x 3 = 151875
```

```
(256, 256, 3)

3

196608
```

In [40]:
```
print("Max: ", np.max(img1_RGB))
print(" ")
print("Min: ", np.min(img1_RGB))
```

```
Max:  255

Min:  0
```

In [41]:
```
print("Max: ", np.max(img2_RGB))
print(" ")
print("Min: ", np.min(img2_RGB))
```

```
Max:  255

Min:  0
```

In [42]:
```
print("Max: ", np.max(img3_RGB))
print(" ")
print("Min: ", np.min(img3_RGB))
```

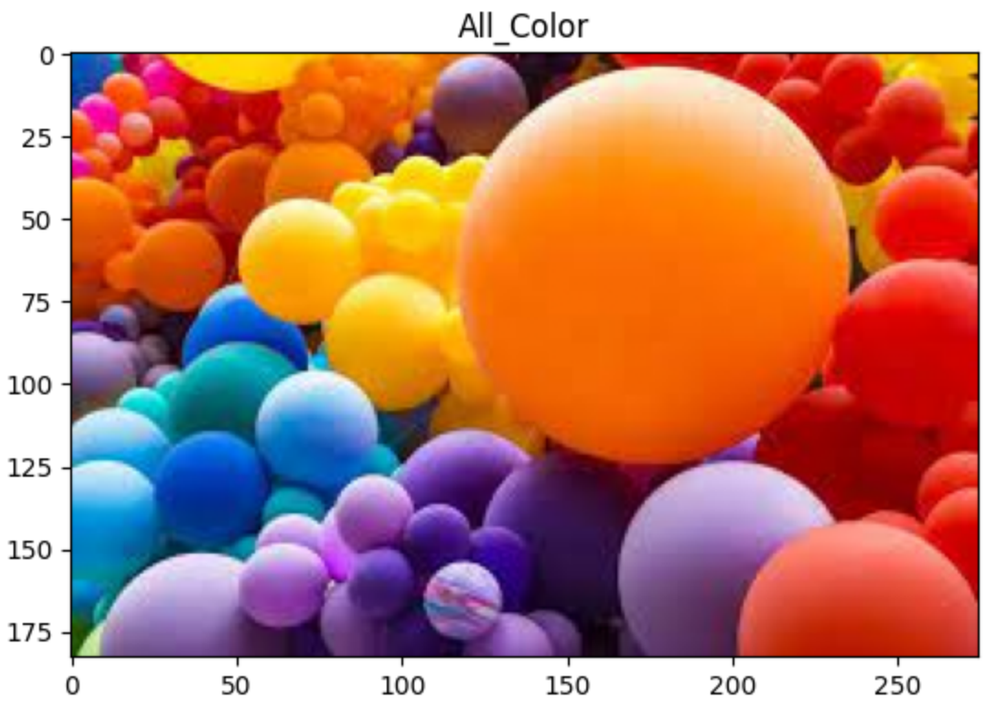
```
Max:  247

Min:  0
```

In [43]:
```
img1_f = img1_RGB.astype(np.float32)/255
```

In [44]:
```
print(type(img1_f))
print(" ")
print(img1_f.shape)
print(" ")
print(img1_f.ndim)
print(" ")
print(img1_f.size)
print(" ")
print(img1_f.dtype)
```

```
<class 'numpy.ndarray'>

(183, 275, 3)

3

150975

float32
```

In [49]:
```
plt.imshow(img1_f)
plt.title('All_Color')
plt.show()
```

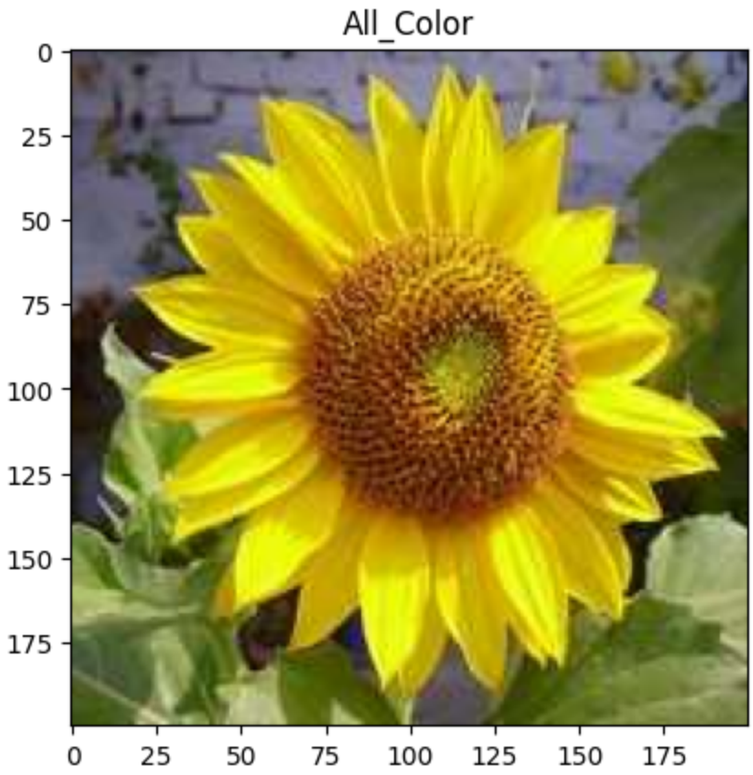


In [45]:
```
img2_f = img2_RGB.astype(np.float32)/255
```

In [46]:
```
print(type(img2_f))
print(" ")
print(img2_f.shape)
print(" ")
print(img2_f.ndim)
print(" ")
print(img2_f.size)
print(" ")
print(img2_f.dtype)
```

```
<class 'numpy.ndarray'>

(200, 200, 3)

3

120000

float32
```

In [54]:
```
plt.imshow(img2_f)
plt.title('All_Color')
plt.show()
```

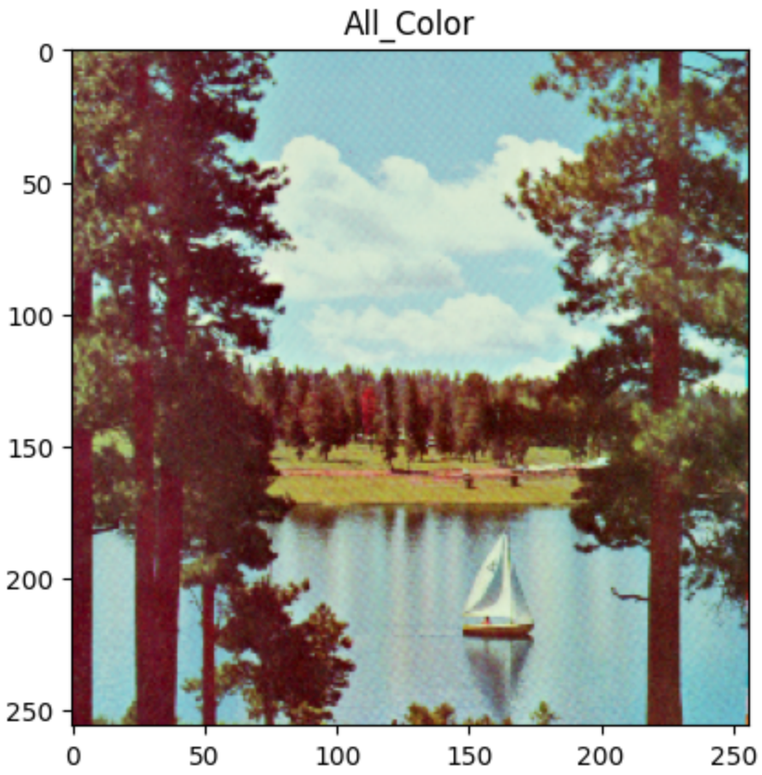


In [47]:
```
img3_f = img3_RGB.astype(np.float32)/255
```

In [48]:
```
print(type(img3_f))
print(" ")
print(img3_f.shape)
print(" ")
print(img3_f.ndim)
print(" ")
print(img3_f.size)
print(" ")
print(img3_f.dtype)
```

```
<class 'numpy.ndarray'>

(256, 256, 3)

3

196608

float32
```

In [55]:
```
plt.imshow(img3_f)
plt.title('All_Color')
plt.show()
```



In [56]:
```
print(np.max(img1_f))
print(" ")
print(np.min(img1_f))
```

```
1.0

0.0
```

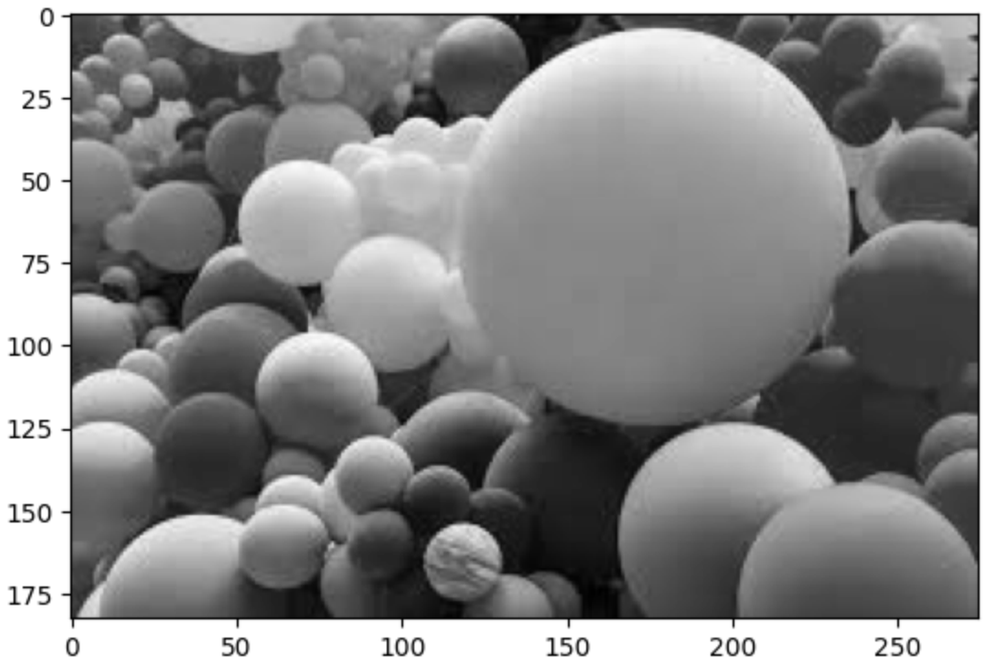
In [57]:
```
print(np.max(img2_f))
print(" ")
print(np.min(img2_f))
```

```
1.0

0.0
```

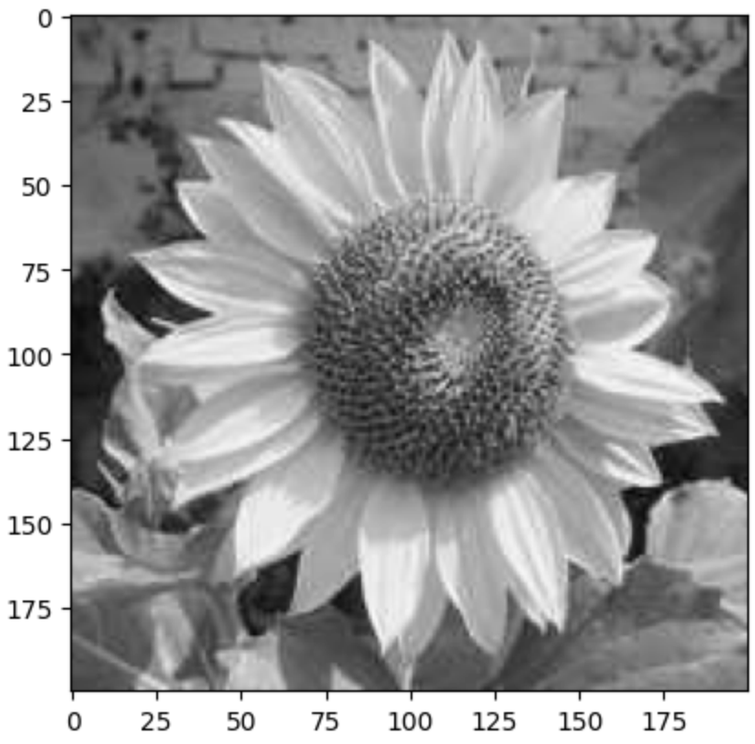
In [58]:
```
print(np.max(img3_f))
print(" ")
print(np.min(img3_f))
```

```
0.96862745

0.0
```

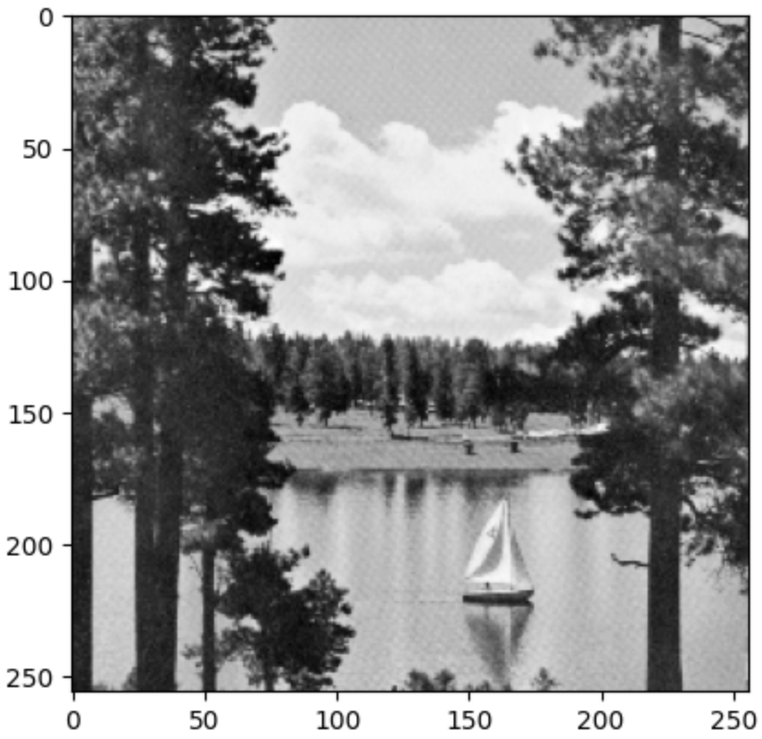
In [63]:
```
# Convert to gray Scale img1
img1_gray_c = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
plt.imshow(img1_gray_c, cmap = 'gray')
plt.show()
```



In [64]:
```
# Convert to gray Scale img2
img2_gray_c = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
plt.imshow(img2_gray_c, cmap = 'gray')
plt.show()
```

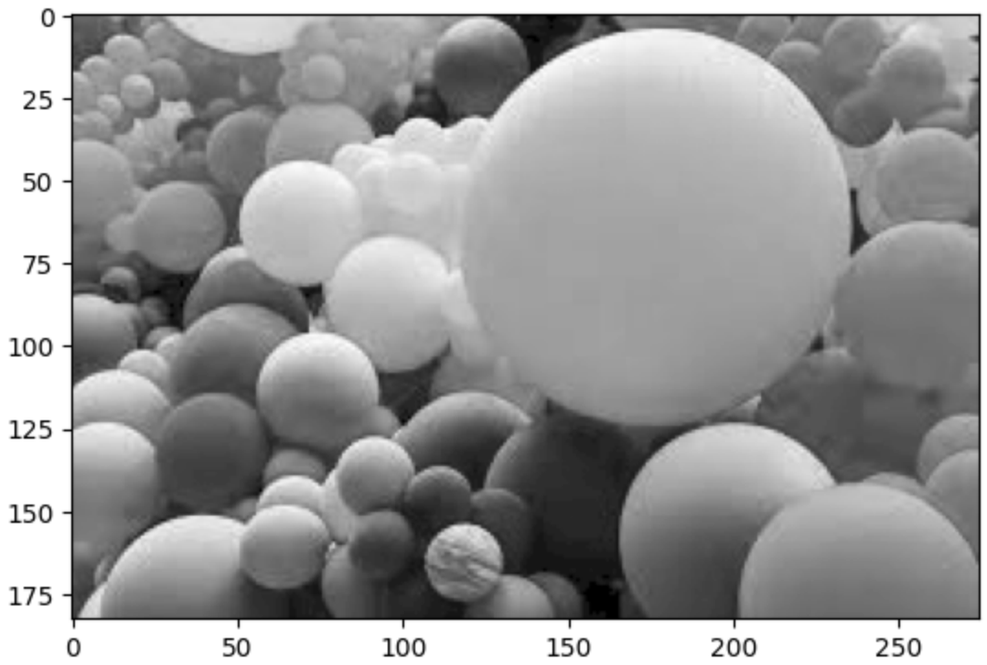


In [65]:
```
# Convert to gray Scale img3
img3_gray_c = cv2.cvtColor(img3, cv2.COLOR_BGR2GRAY)
plt.imshow(img3_gray_c, cmap = 'gray')
plt.show()
```



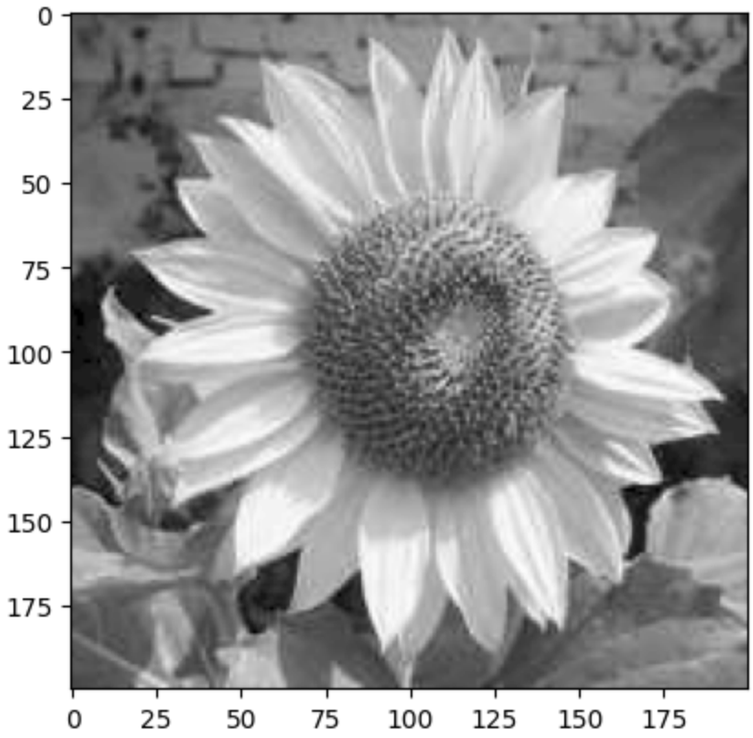
In [62]:
```
# Read the image directly in a gray scale

img1_gray = cv2.imread('/content/drive/MyDrive/Image Processing Humber/AllColo

plt.imshow(img1_gray, cmap = 'gray')
```

Out[62]:
```
<matplotlib.image.AxesImage at 0x78d4748738b0>
```



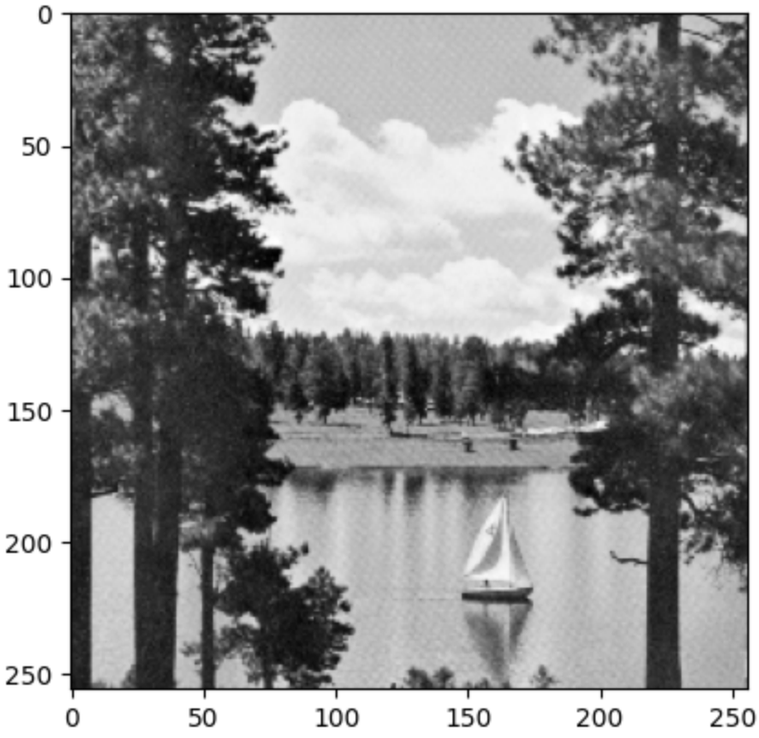
In [61]:
```
# Read the image directly in a gray scale

img2_gray = cv2.imread('/content/drive/MyDrive/Image Processing Humber/SunFlow

plt.imshow(img2_gray, cmap = 'gray')
```

Out[61]:
```
<matplotlib.image.AxesImage at 0x78d474800340>
```



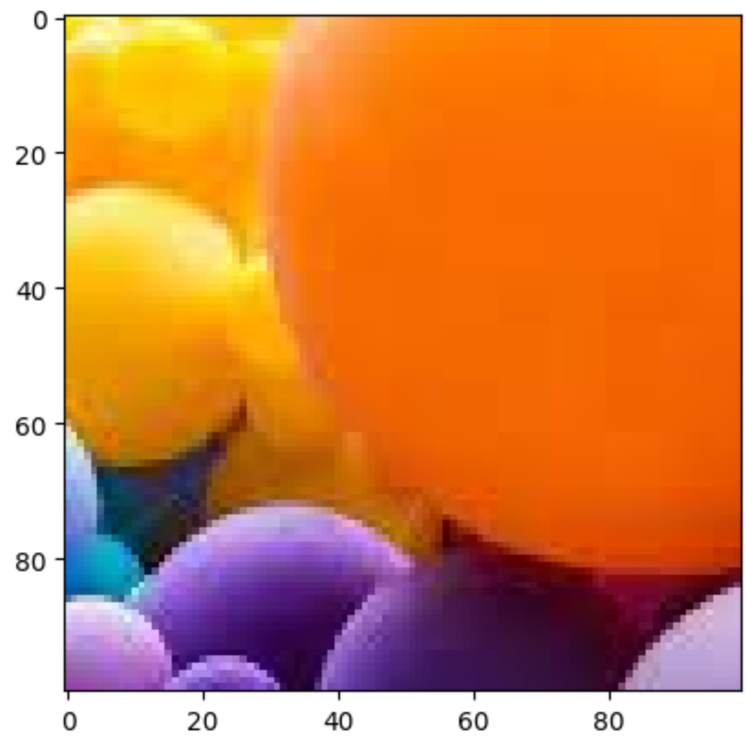
In [60]:
```
# Read the image directly in a gray scale

img3_gray = cv2.imread('/content/drive/MyDrive/Image Processing Humber/boatcol

plt.imshow(img3_gray, cmap = 'gray')
```

Out[60]:
```
<matplotlib.image.AxesImage at 0x78d4749731f0>
```



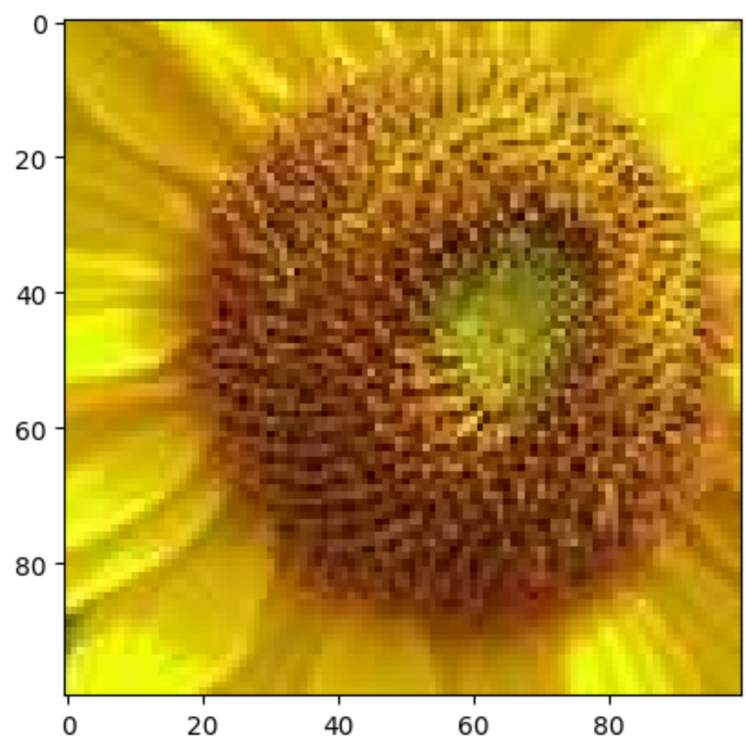
In [79]:
```
# Crop Images 100 x 100 Image 1

img1_crop = img1_f[42:142,88:188, :]
plt.imshow(img1_crop)
```

Out[79]: <matplotlib.image.AxesImage at 0x78d47458ae60>



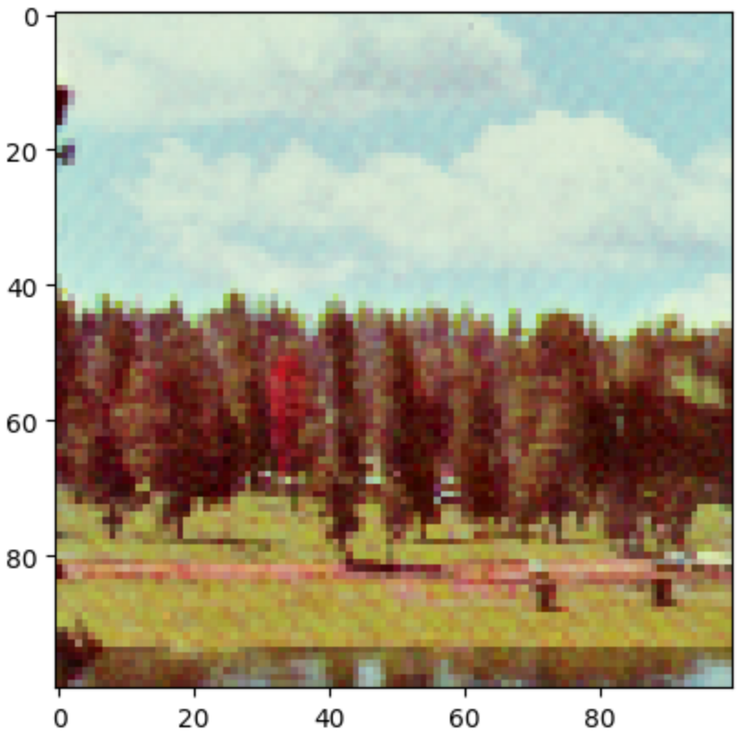
In [80]:
```
# Crop Images 100 x 100 Image 3

img2_crop = img2_f[50:150,50:150, :]
plt.imshow(img2_crop)
```

Out[80]: <matplotlib.image.AxesImage at 0x78d4743f90c0>

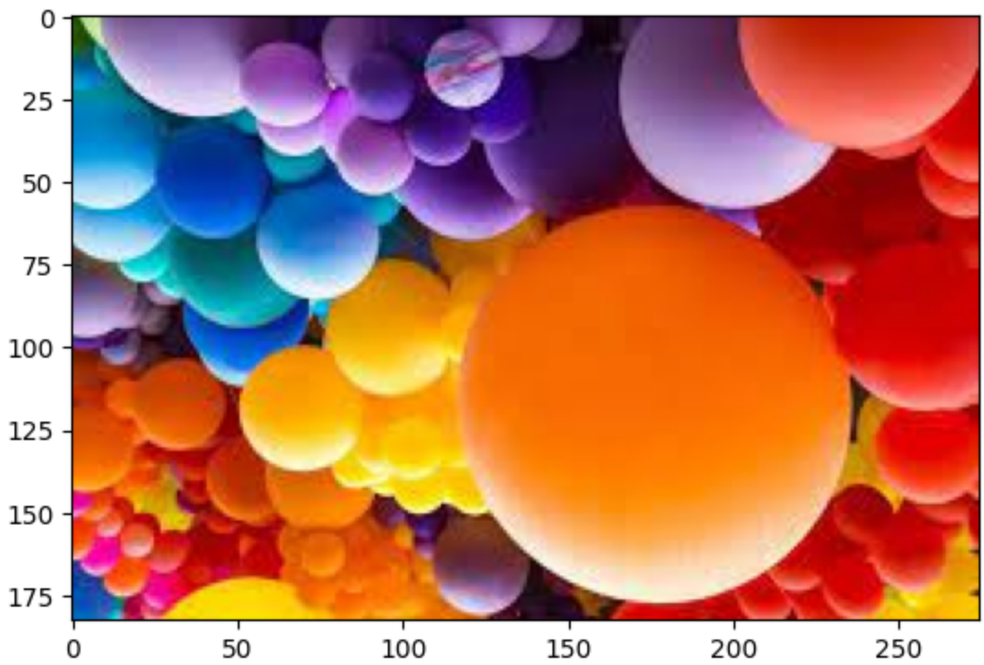


In [81]:
```
# Crop Images 100 x 100 Image 3

img3_crop = img3_f[78:178,78:178, :]
plt.imshow(img3_crop)
```

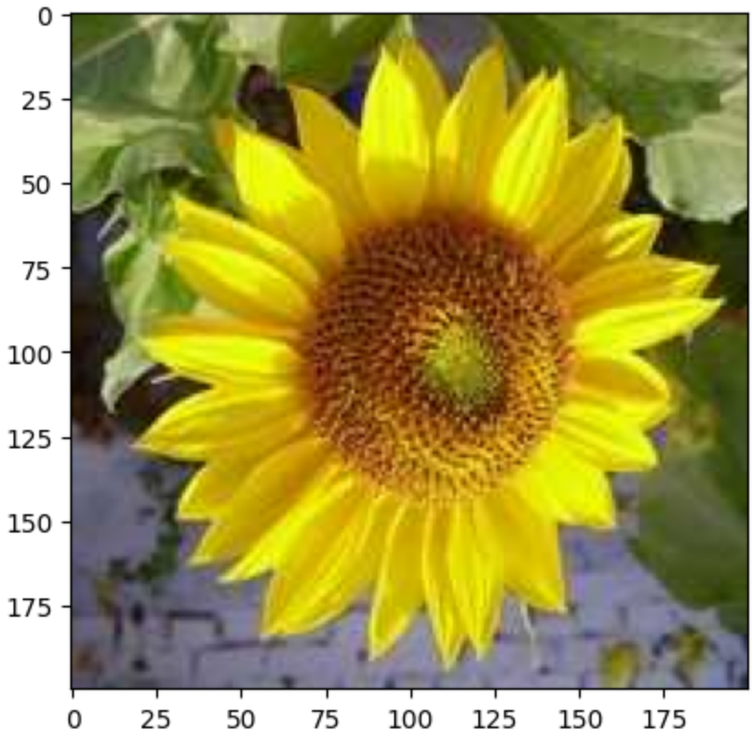
Out[81]: <matplotlib.image.AxesImage at 0x78d474462530>



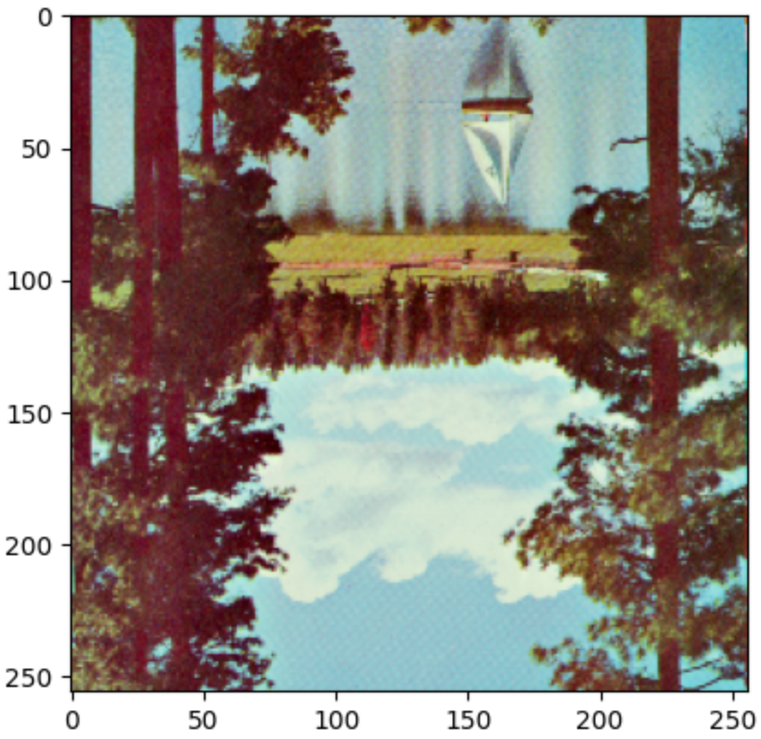
In [82]:
```
# Image Flipping along X-axis Image 1

img1_flipX = cv2.flip(img1_RGB,0)
plt.imshow(img1_flipX)
plt.show()
```



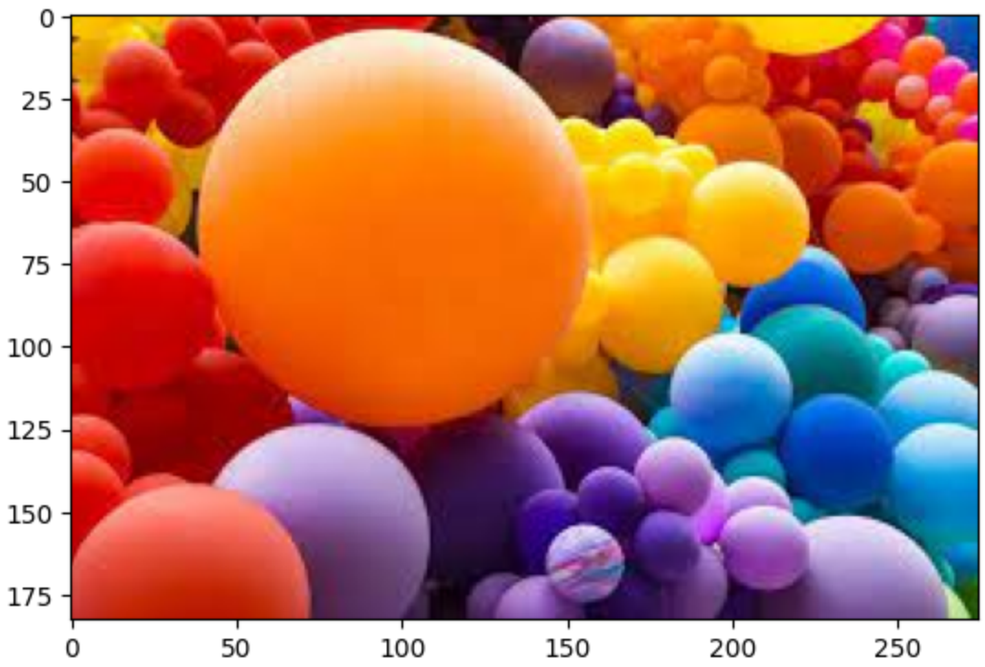
In [83]:
```
# Image Flipping along X-axis Image 2

img2_flipX = cv2.flip(img2_RGB,0)
plt.imshow(img2_flipX)
plt.show()
```



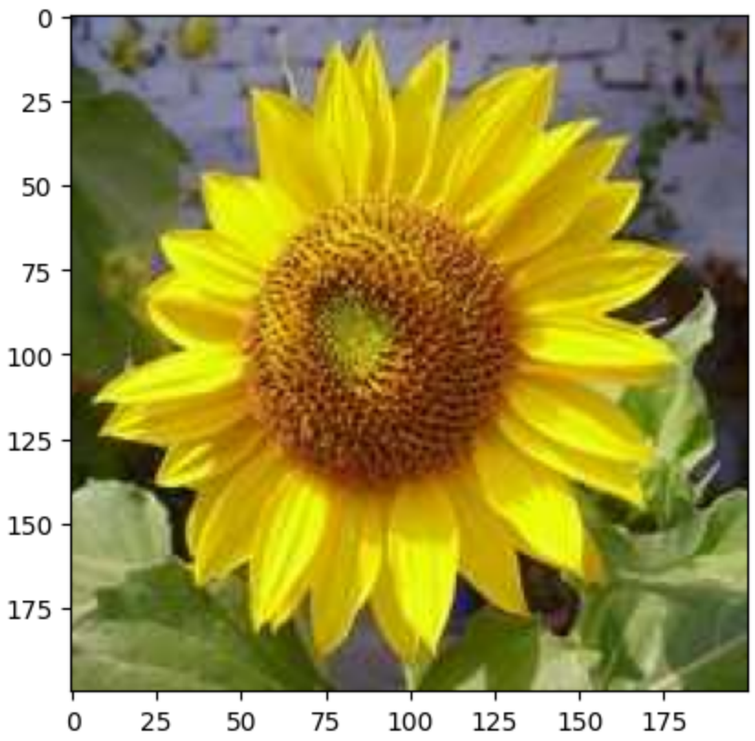
In [84]:
```
# Image Flipping along X-axis Image 3

img3_flipX = cv2.flip(img3_RGB,0)
plt.imshow(img3_flipX)
plt.show()
```



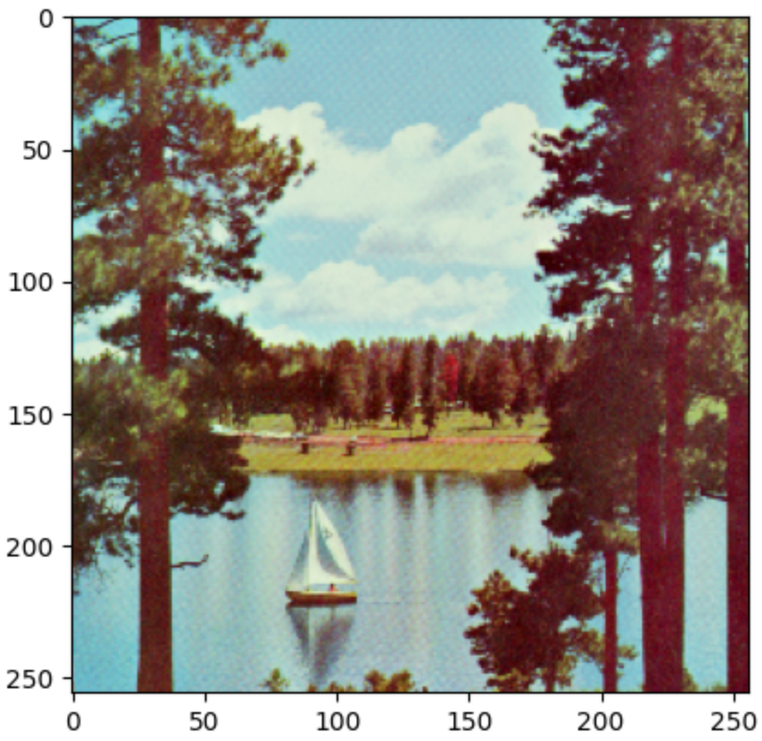
In [87]:
```
# Image Flipping along X-axis Image 1

img1_flipY = cv2.flip(img1_RGB,1)
plt.imshow(img1_flipY)
plt.show()
```



In [88]:
```
# Image Flipping along X-axis Image 2

img2_flipY = cv2.flip(img2_RGB,1)
plt.imshow(img2_flipY)
plt.show()
```

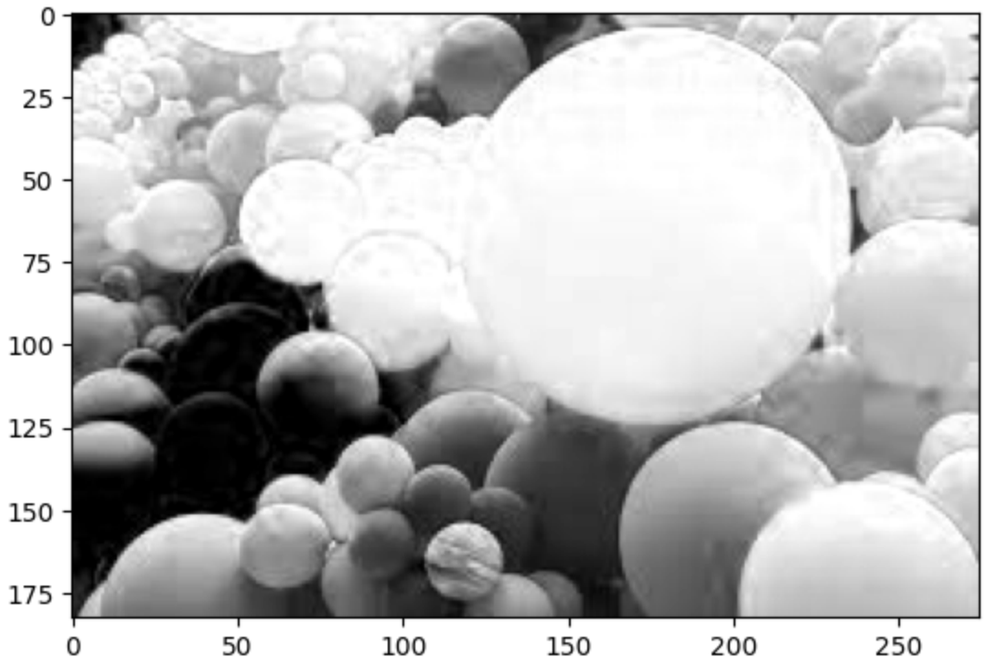


In [89]:
```
# Image Flipping along X-axis Image 3

img3_flipY = cv2.flip(img3_RGB,1)
plt.imshow(img3_flipY)
plt.show()
```



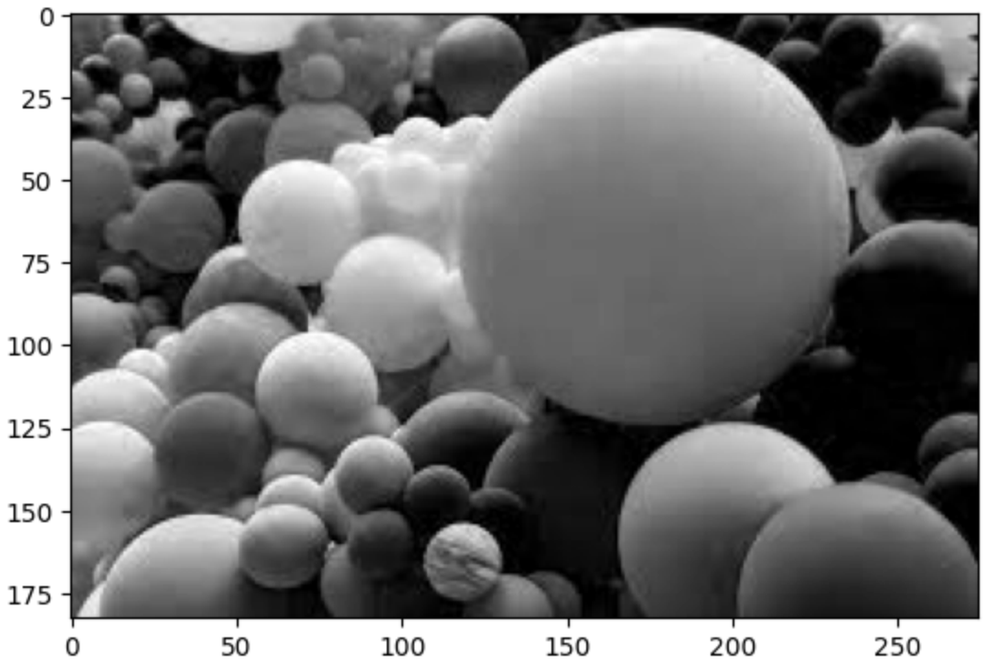
In [129]:
```
img1_r = img1_RGB[:,:,0]
plt.imshow(img1_r, cmap = 'gray') # see the red
```

Out[129]:
```
<matplotlib.image.AxesImage at 0x78d47330f7f0>
```



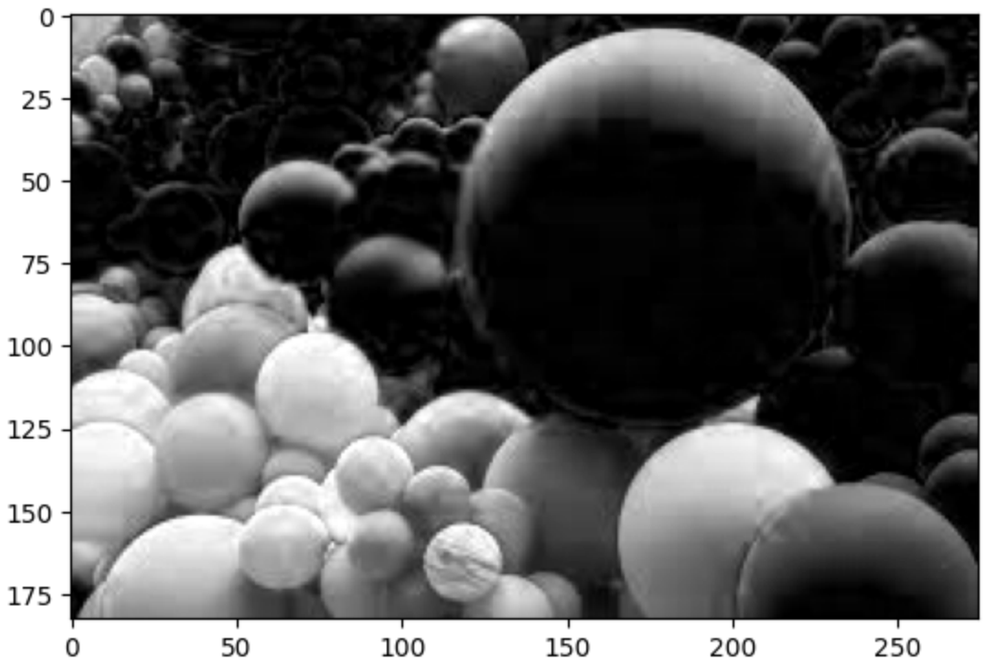
In [130]:
```
img1_g = img1_RGB[:,:,1]
plt.imshow(img1_g, cmap = 'gray') # see the green
```

Out[130]: <matplotlib.image.AxesImage at 0x78d473386b30>



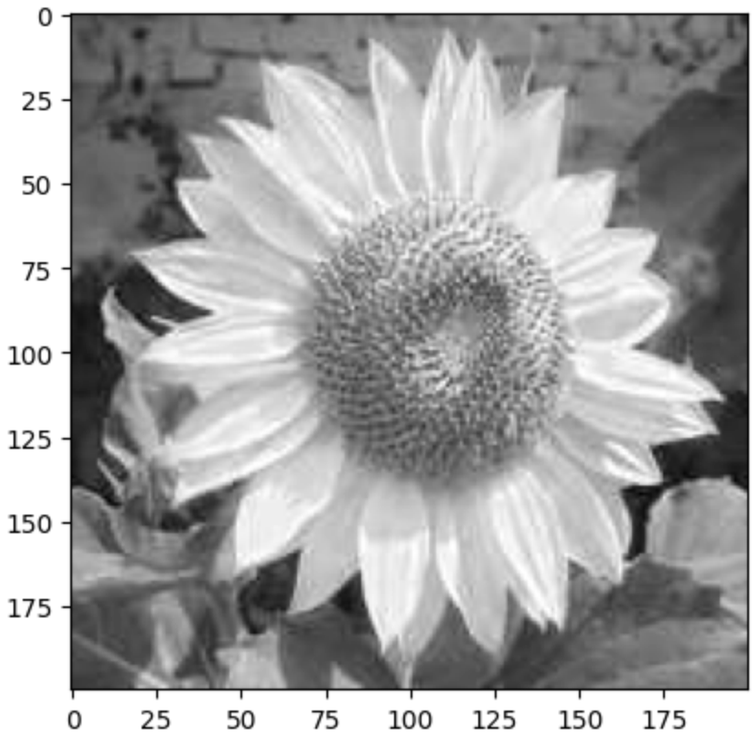
In [131]:
```
img1_b = img1_RGB[:,:,2]
plt.imshow(img1_b, cmap = 'gray') # see the blue
```

Out[131]: <matplotlib.image.AxesImage at 0x78d473201f90>



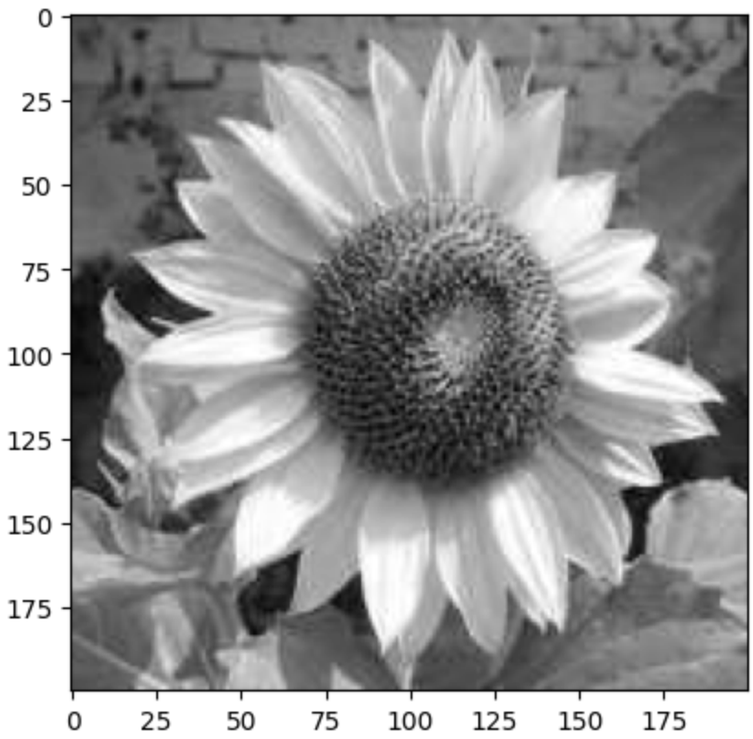
In [132]:
```
img2_r = img2_RGB[:,:,0]
plt.imshow(img2_r, cmap = 'gray') # see the red
```

Out[132]: <matplotlib.image.AxesImage at 0x78d47433a980>



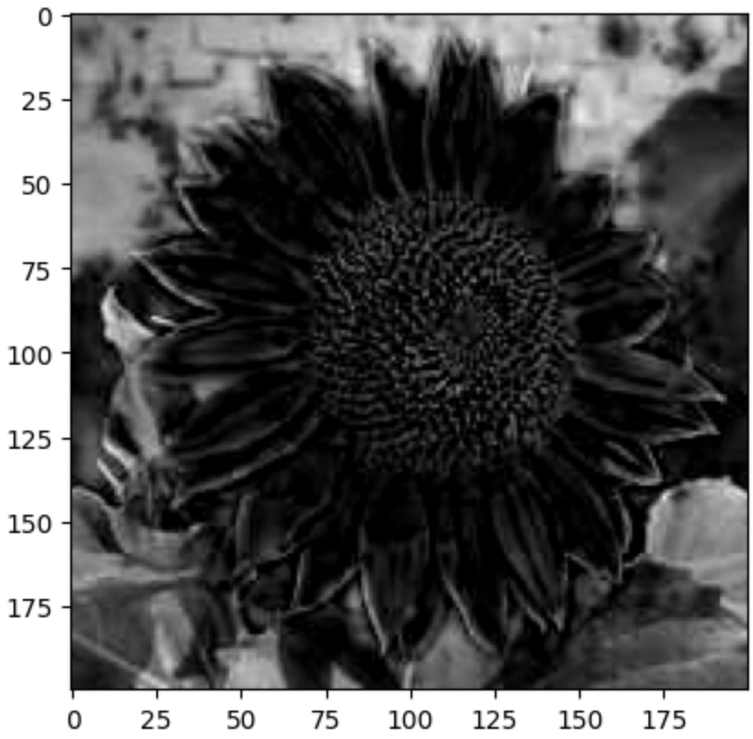
In [133]:
```
img2_g = img2_RGB[:,:,1]
plt.imshow(img2_g, cmap = 'gray') # see the green
```

Out[133]: <matplotlib.image.AxesImage at 0x78d47471a3b0>



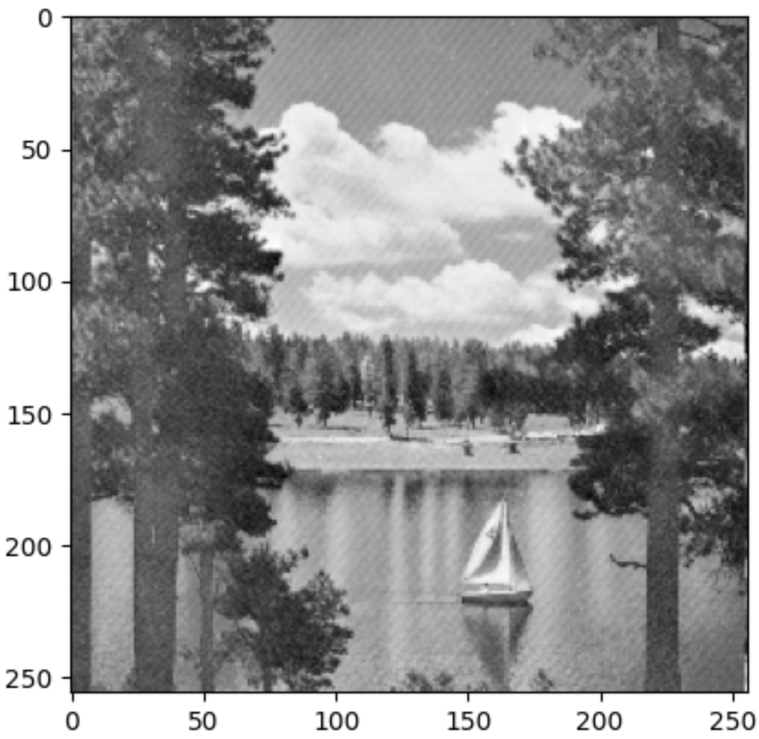
In [134]:
```
img2_b = img2_RGB[:,:,2]
plt.imshow(img2_b, cmap = 'gray') # see the blue
```

Out[134]: <matplotlib.image.AxesImage at 0x78d473114430>



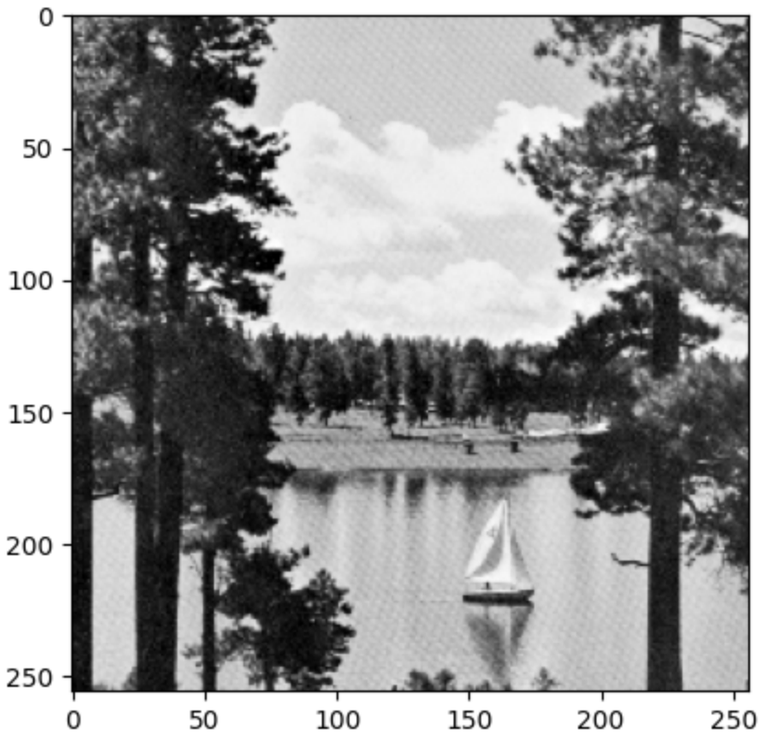
In [135]:
```
img3_r = img3_RGB[:,:,0]
plt.imshow(img3_r, cmap = 'gray') # see the red
```

Out[135]: <matplotlib.image.AxesImage at 0x78d4731839a0>



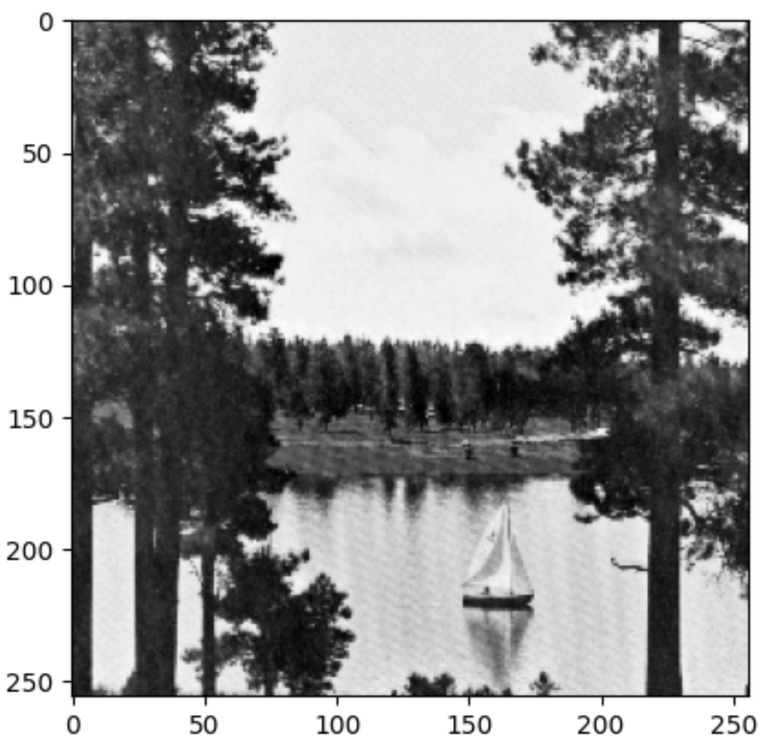
In [136]:
```
img3_g = img3_RGB[:,:,1]
plt.imshow(img3_g, cmap = 'gray') # see the green
```

Out[136]: <matplotlib.image.AxesImage at 0x78d4730046d0>



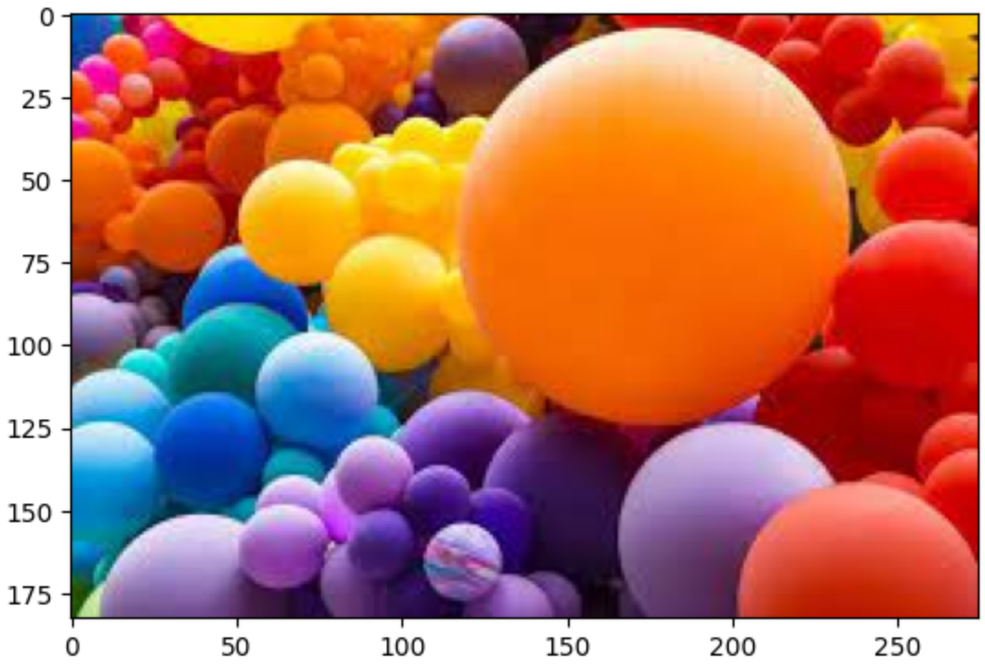
In [137]:
```
img3_b = img3_RGB[:,:,2]
plt.imshow(img3_b, cmap = 'gray') # see the blue
```

Out[137]: <matplotlib.image.AxesImage at 0x78d47307d6c0>



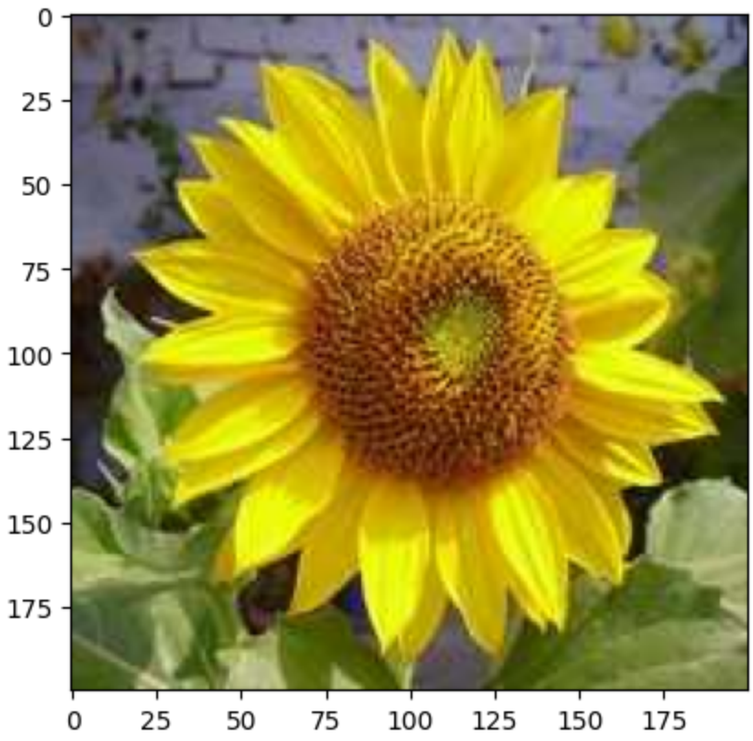
In [139]:
```
image_merge = cv2.merge([img1_r, img1_g, img1_b])
plt.imshow(image_merge)
```

Out[139]: <matplotlib.image.AxesImage at 0x78d472f80490>



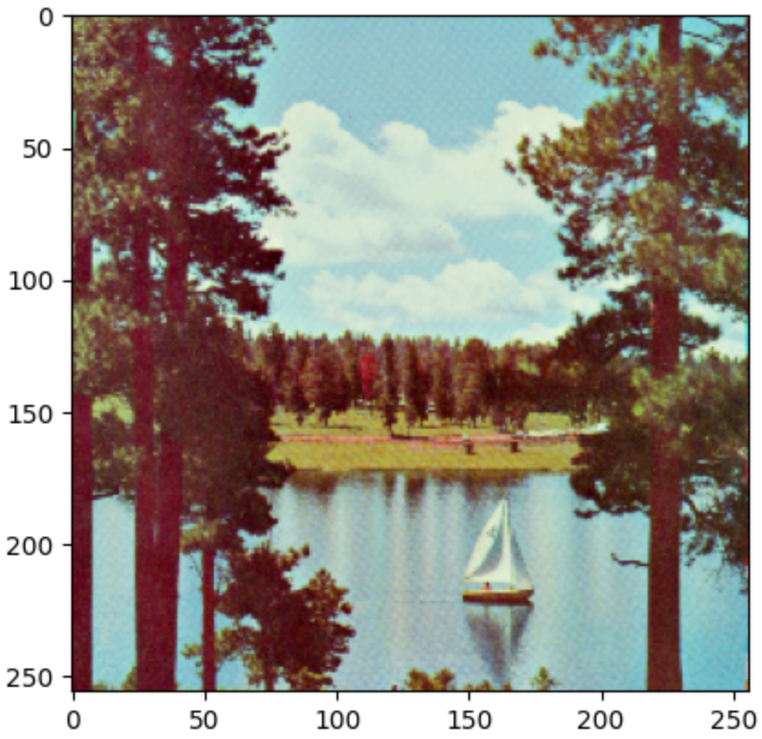
In [140]:
```
image2_merge = cv2.merge([img2_r, img2_g, img2_b])
plt.imshow(image2_merge)
```

Out[140]: <matplotlib.image.AxesImage at 0x78d472de2a10>



In [141]:
```
image3_merge = cv2.merge([img3_r, img3_g, img3_b])
plt.imshow(image3_merge)
```

Out[141]: <matplotlib.image.AxesImage at 0x78d472e699c0>



In [ ]: