In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imshow, imread
from skimage.color import rgb2hsv, hsv2rgb
import cv2
```

In [2]:
```
image = imread(r'C:\Users\hp\Downloads/tree.jpg')
plt.figure(num=None, figsize=(8, 6), dpi=80)
imshow(image);
```

In [6]:
```
from PIL import Image
im = Image.open(r'C:\Users\hp\Downloads/tree.jpg', 'r')
width, height = im.size
pixel_values = list(im.getdata())
```

In [8]:
```
import cv2
```

In [9]:
```
flags = [i for i in dir(cv2) if i.startswith('COLOR_')]
```

In [10]:
```
len(flags)
```

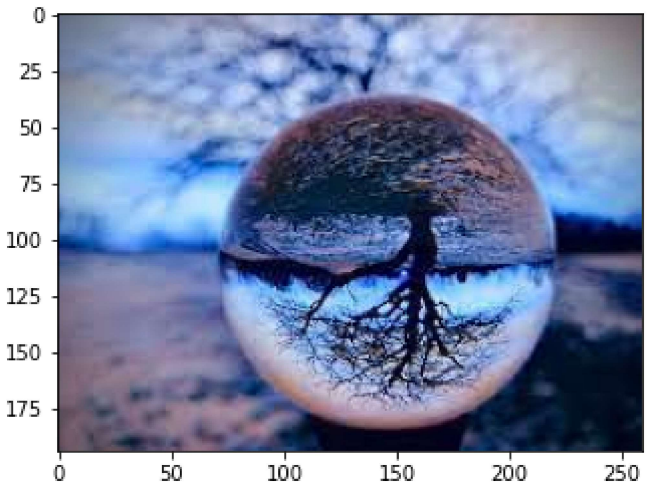
Out[10]: 274

In [11]:
```
flags[40]
```

Out[11]: 'COLOR_BGR2HLS'

In [12]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [14]:
```
image = cv2.imread(r'C:\Users\hp\Downloads/tree.jpg')
plt.imshow(image)
plt.show()
```



In [15]:
```
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)
plt.show()
```

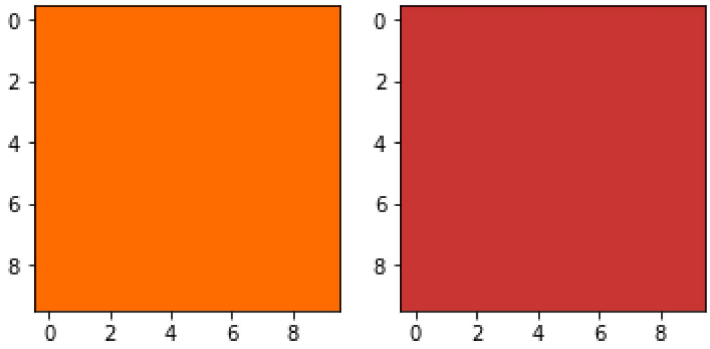
In [16]:
```
hsv_image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
```

In [22]:
```
from matplotlib.colors import hsv_to_rgb
```

In [23]:
```
lo_square = np.full((10, 10, 3), light_orange, dtype=np.uint8) / 255.0
do_square = np.full((10, 10, 3), dark_orange, dtype=np.uint8) / 255.0
```

In [24]:
```
plt.subplot(1, 2, 1)
plt.imshow(hsv_to_rgb(do_square))
plt.subplot(1, 2, 2)
```

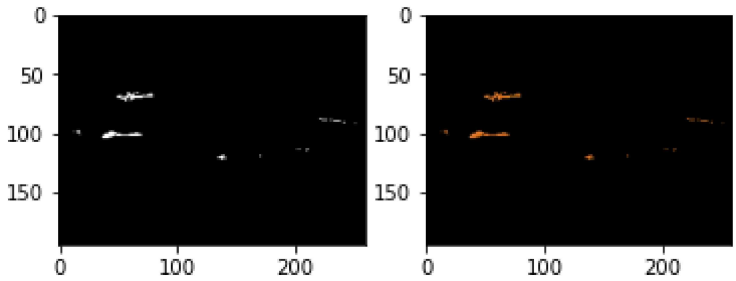
```
plt.imshow(hsv_to_rgb(lo_square))
plt.show()
```



In [25]:
```
mask = cv2.inRange(hsv_image, light_orange, dark_orange)
```

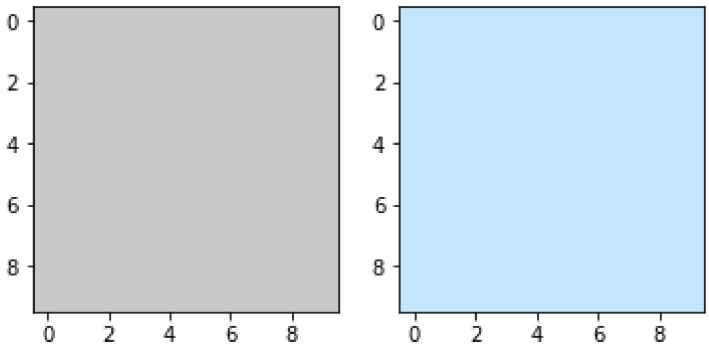
In [26]:
```
result = cv2.bitwise_and(image, image, mask=mask)
```

In [27]:
```
plt.subplot(1, 2, 1)
plt.imshow(mask, cmap="gray")
plt.subplot(1, 2, 2)
plt.imshow(result)
plt.show()
```



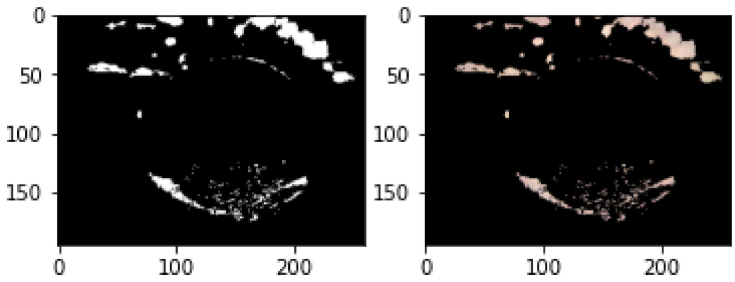
In [28]:
```
light_white = (0, 0, 200)
dark_white = (145, 60, 255)
```

In [29]:
```
lw_square = np.full((10, 10, 3), light_white, dtype=np.uint8) / 255.0
dw_square = np.full((10, 10, 3), dark_white, dtype=np.uint8) / 255.0

plt.subplot(1, 2, 1)
plt.imshow(hsv_to_rgb(lw_square))
plt.subplot(1, 2, 2)
plt.imshow(hsv_to_rgb(dw_square))
plt.show()
```



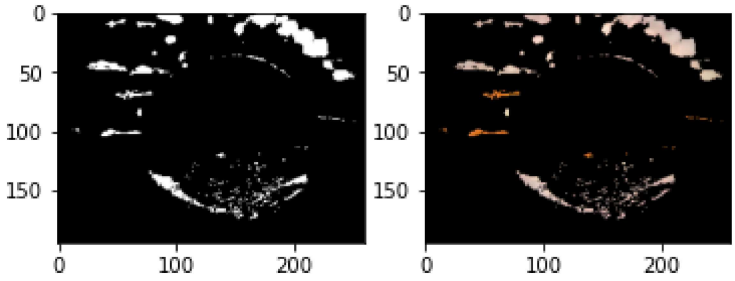
In [31]:

```
mask_white = cv2.inRange(hsv_image, light_white, dark_white)
result_white = cv2.bitwise_and(image, image, mask=mask_white)

plt.subplot(1, 2, 1)
plt.imshow(mask_white, cmap="gray")
plt.subplot(1, 2, 2)
plt.imshow(result_white)
plt.show()
```



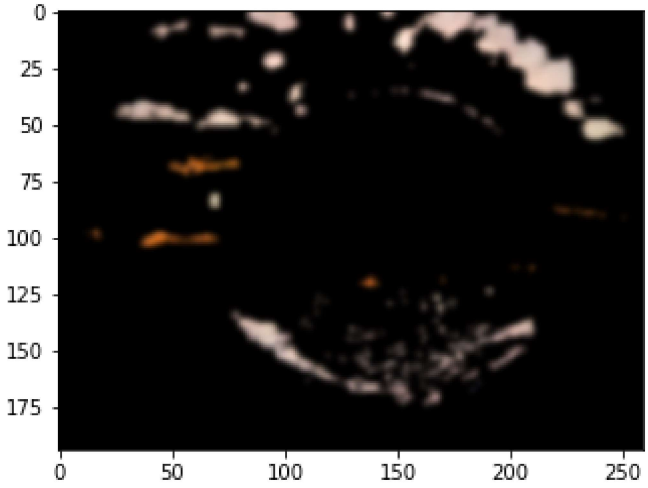
In [32]:

```
final_mask = mask + mask_white

final_result = cv2.bitwise_and(image, image, mask=final_mask)
plt.subplot(1, 2, 1)
plt.imshow(final_mask, cmap="gray")
plt.subplot(1, 2, 2)
plt.imshow(final_result)
plt.show()
```



In [33]:

```
blur = cv2.GaussianBlur(final_result, (7, 7), 0)
plt.imshow(blur)
plt.show()
```


0
25
50
75
100
125
150
175
0
50
100
150
200
250