

```
In [15]: # import the Python Image
# processing Library
from PIL import Image

# Giving The Original image Directory
# Specified
Original_Image = Image.open(r'C:/Users/hp/Downloads/tree.jpg')

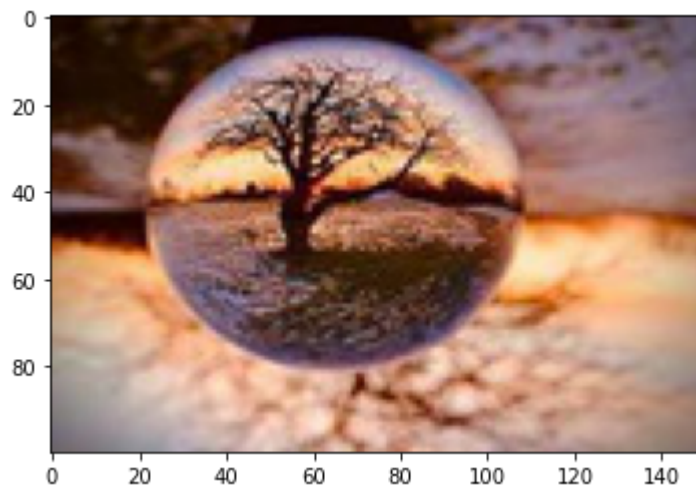
# Rotate Image By 180 Degree
rotated_image1 = Original_Image.rotate(180)

# This is Alternative Syntax To Rotate
# The Image
rotated_image2 = Original_Image.transpose(Image.ROTATE_90)

# This Will Rotate Image By 60 Degree
rotated_image3 = Original_Image.rotate(60)
```

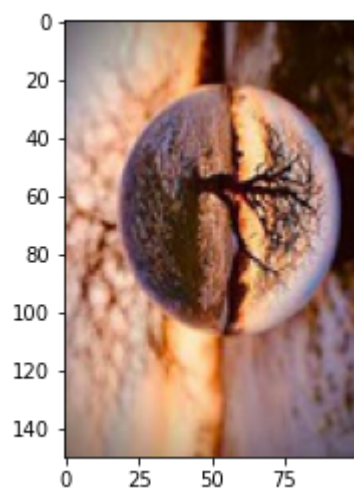
```
In [16]: plt.imshow(rotated_image1)
```

```
Out[16]: <matplotlib.image.AxesImage at 0x205e7e17b50>
```



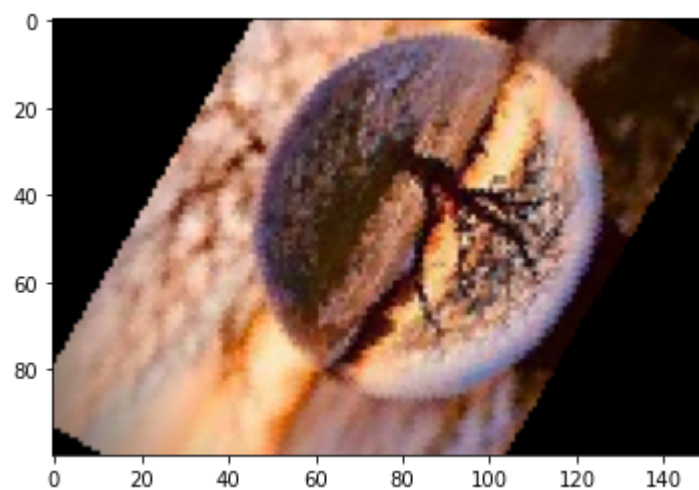
```
In [17]: plt.imshow(rotated_image2)
```

```
Out[17]: <matplotlib.image.AxesImage at 0x205e7e71880>
```



```
In [18]: plt.imshow(rotated_image3)
```

```
Out[18]: <matplotlib.image.AxesImage at 0x205e7ed0280>
```



```
In [23]: # Importing Image class from PIL module
         from PIL import Image

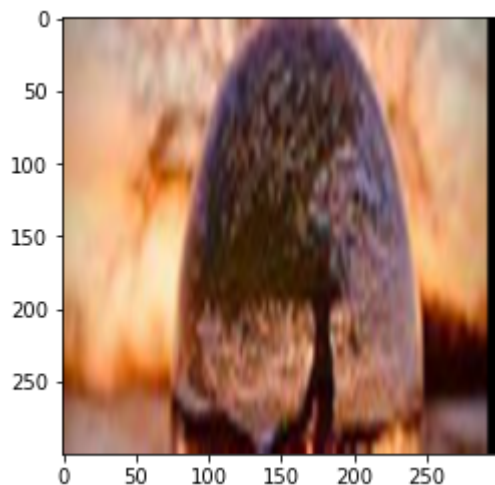
         # Opens a image in RGB mode
         im = Image.open(r"C:\Users\hp\Downloads\tree.jpg")

         # Size of the image in pixels (size of original image)
         # (This is not mandatory)
         width, height = im.size

         # Setting the points for cropped image
         left = 4
         top = height / 5
         right = 154
         bottom = 3 * height / 5

         # Cropped image of above dimension
         # (It will not change original image)
         im1 = im.crop((left, top, right, bottom))
         newsize = (300, 300)
         im1 = im1.resize(newsize)
         # Shows the image in image viewer
         plt.imshow(im1)
```

Out[23]: <matplotlib.image.AxesImage at 0x205e7f8c160>

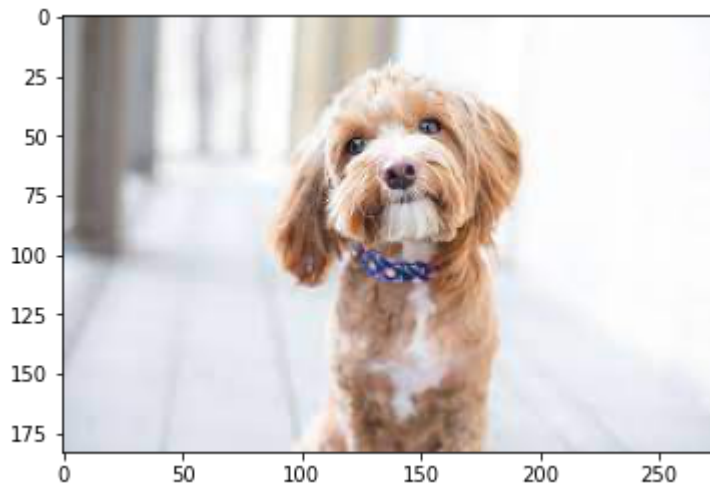


In []:

```
In [30]: from PIL import Image
```

```
image = Image.open(r'C:/Users/hp/Downloads\a.jpg')  
plt.imshow(image)
```

```
Out[30]: <matplotlib.image.AxesImage at 0x205e7fe60a0>
```



```
In [31]: # The file format of the source file.
```

```
print(image.format) # Output: JPG
```

```
# The pixel format used by the image. Typical values are "1", "L", "RGB", or "CMY"
```

```
print(image.mode) # Output: RGB
```

```
# Image size, in pixels. The size is given as a 2-tuple (width, height).
```

```
print(image.size) # Output: (1920, 1280)
```

```
# Colour palette table, if any.
```

```
print(image.palette) # Output: None
```

JPEG

RGB

(275, 183)

None

```
In [37]: image = Image.open(r'C:/Users/hp/Downloads\a.jpg')
```

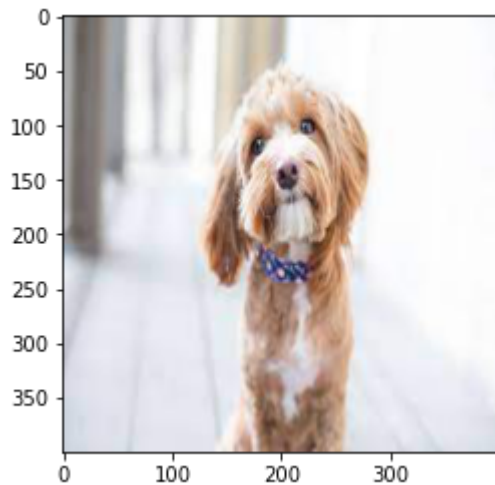
```
image.save('new_image.png')
```

```
In [41]: image = Image.open(r'C:/Users/hp/Downloads\a.jpg')
new_image = image.resize((400, 400))
new_image.save('image_400.jpg')

print(image.size) # Output: (1920, 1280)
print(new_image.size) # Output: (400, 400)
plt.imshow(new_image)
```

```
(275, 183)
(400, 400)
```

Out[41]: <matplotlib.image.AxesImage at 0x205e8040820>

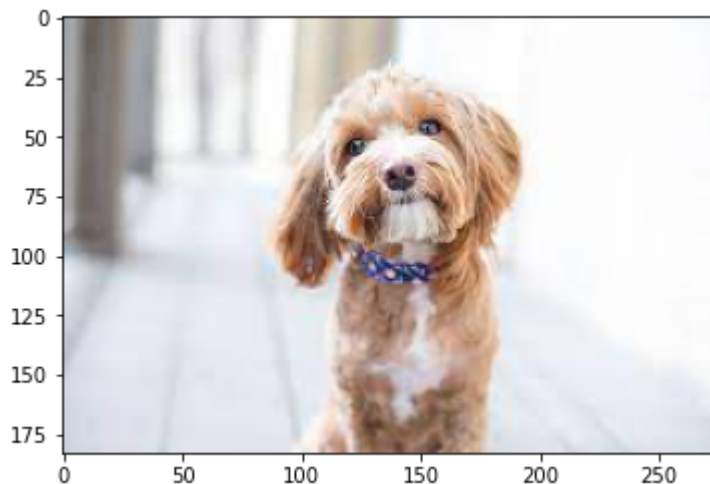


```
In [42]: image = Image.open(r'C:/Users/hp/Downloads\a.jpg')
image.thumbnail((400, 400))
image.save('image_thumbnail.jpg')

print(image.size) # Output: (400, 267)
plt.imshow(image)
```

```
(275, 183)
```

Out[42]: <matplotlib.image.AxesImage at 0x205e7df9880>



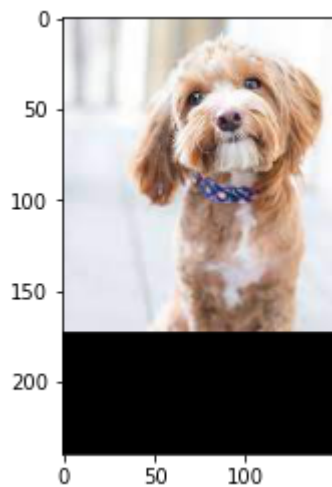
In []:

```
In [61]: image = Image.open(r'C:/Users/hp/Downloads\a.jpg')
box = (50, 10, 200, 250)
cropped_image = image.crop(box)
cropped_image.save('cropped_image.jpg')

# Print size of cropped image
print(cropped_image.size) # Output: (150, 240)
plt.imshow(cropped_image)
```

(150, 240)

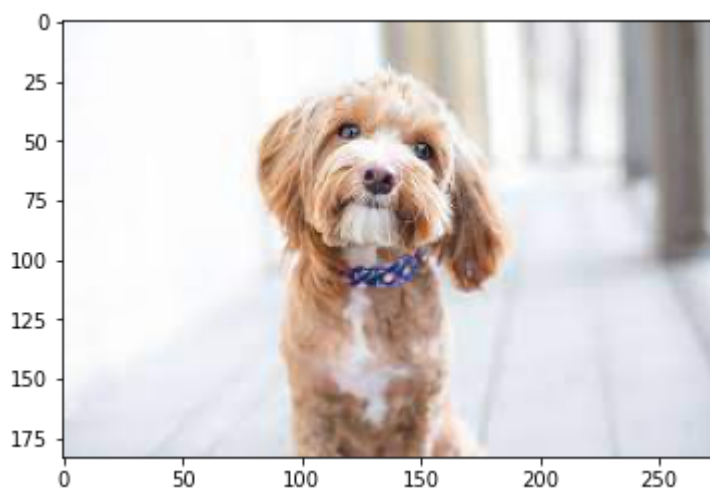
Out[61]: <matplotlib.image.AxesImage at 0x205e9669bb0>



```
In [63]: image = Image.open(r'C:/Users/hp/Downloads\a.jpg')

image_flip = image.transpose(Image.FLIP_LEFT_RIGHT)
image_flip.save('image_flip.jpg')
plt.imshow(image_flip)
```

Out[63]: <matplotlib.image.AxesImage at 0x205e96c1a90>



```
In [70]: image = Image.open(r'C:/Users/hp/Downloads\a.jpg')

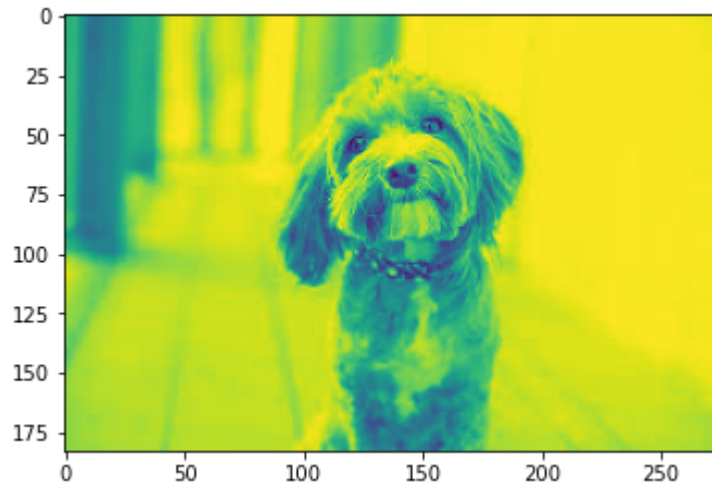
greyscale_image = image.convert('L')
greyscale_image.save('greyscale_image.jpg')

print(image.mode) # Output: RGB
print(greyscale_image.mode) # Output: L
plt.imshow(greyscale_image)
```

RGB

L

```
Out[70]: <matplotlib.image.AxesImage at 0x205e9984d60>
```



```
In [72]: image = Image.open(r'C:/Users/hp/Downloads\a.jpg')
red, green, blue = image.split()

print(image.mode) # Output: RGB
print(red.mode) # Output: L
print(green.mode) # Output: L
print(blue.mode) # Output: L

new_image = Image.merge("RGB", (green, red, blue))
new_image.save('new_image.jpg')

print(new_image.mode) # Output: RGB
plt.imshow(new_image)
```

RGB

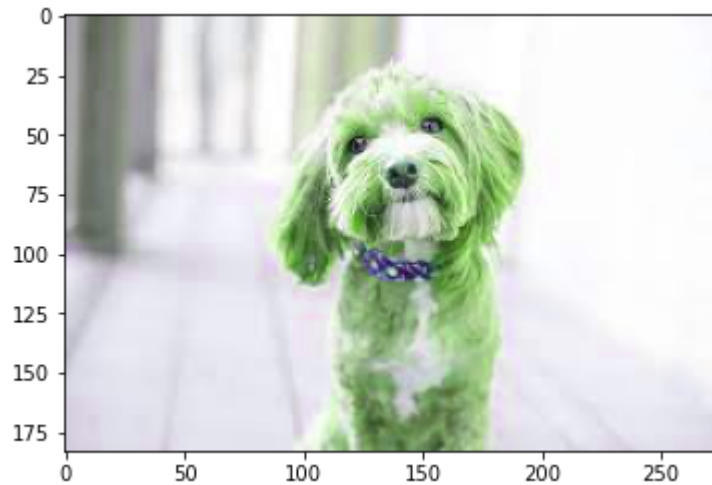
L

L

L

RGB

Out[72]: <matplotlib.image.AxesImage at 0x205ea9bb4f0>

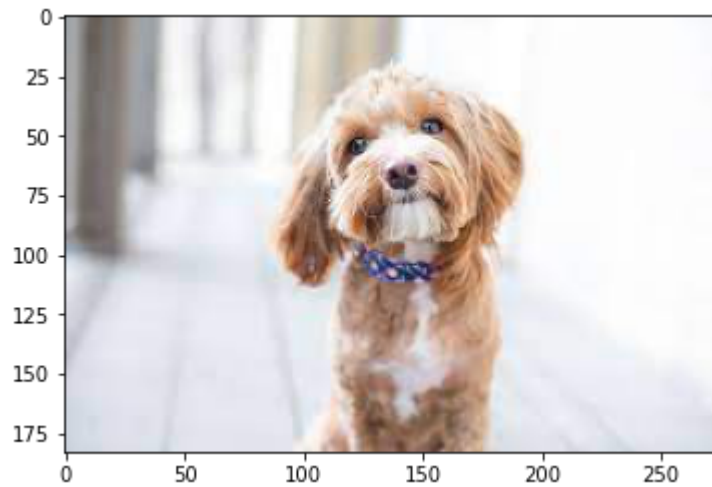



```
In [73]: from PIL import Image, ImageEnhance

image = Image.open(r'C:/Users/hp/Downloads\a.jpg')

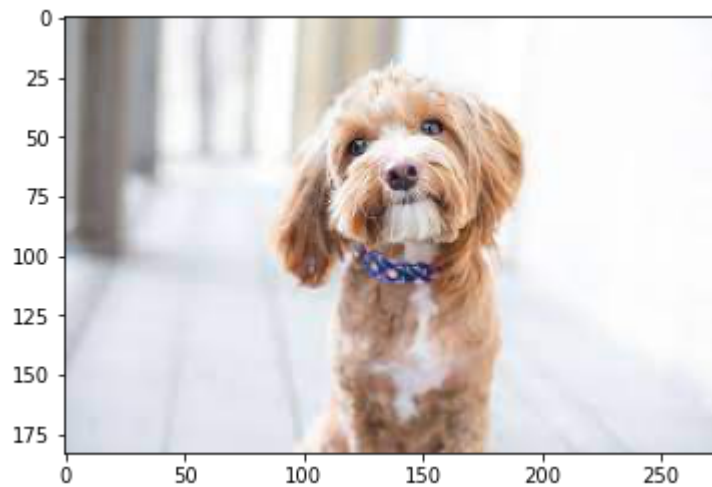
contrast = ImageEnhance.Contrast(image)
contrast.enhance(1.5).save('contrast.jpg')
plt.imshow(image)
```

Out[73]: <matplotlib.image.AxesImage at 0x205e942f940>



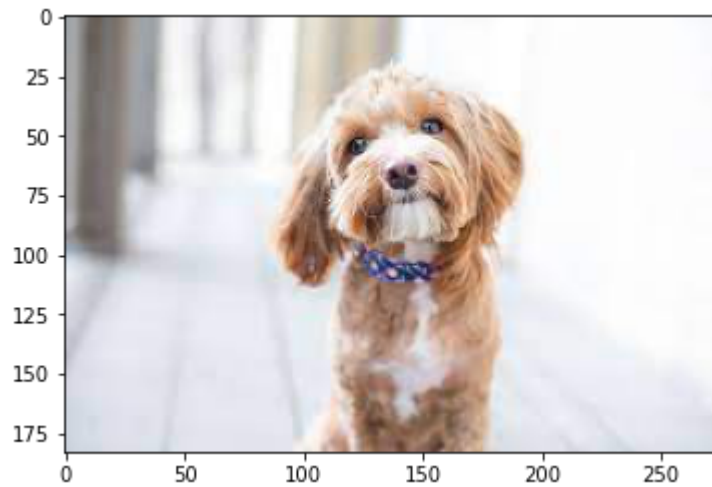
```
In [75]: color = ImageEnhance.Color(image)
color.enhance(1.5).save('color.jpg')
plt.imshow(image)
```

Out[75]: <matplotlib.image.AxesImage at 0x205eaa5f0a0>



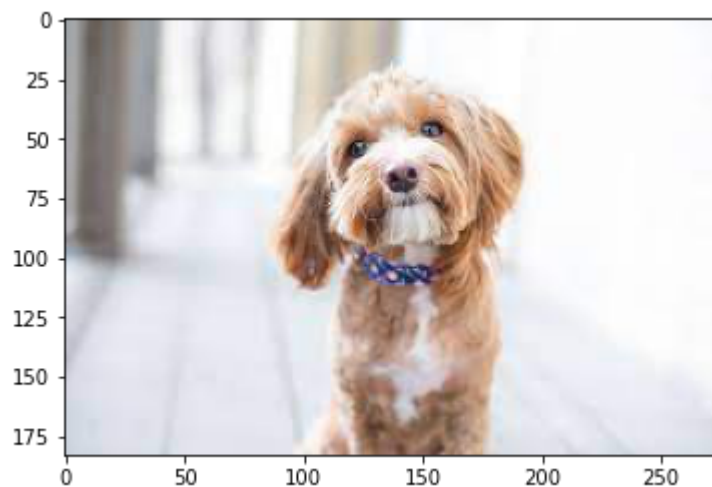
```
In [77]: brightness = ImageEnhance.Brightness(image)
brightness.enhance(2.5).save('brightness.jpg')
plt.imshow(image)
```

Out[77]: <matplotlib.image.AxesImage at 0x205eab14460>



```
In [78]: sharpness = ImageEnhance.Sharpness(image)
sharpness.enhance(1.5).save('sharpness.jpg')
plt.imshow(image)
```

Out[78]: <matplotlib.image.AxesImage at 0x205eab6e640>



In []:

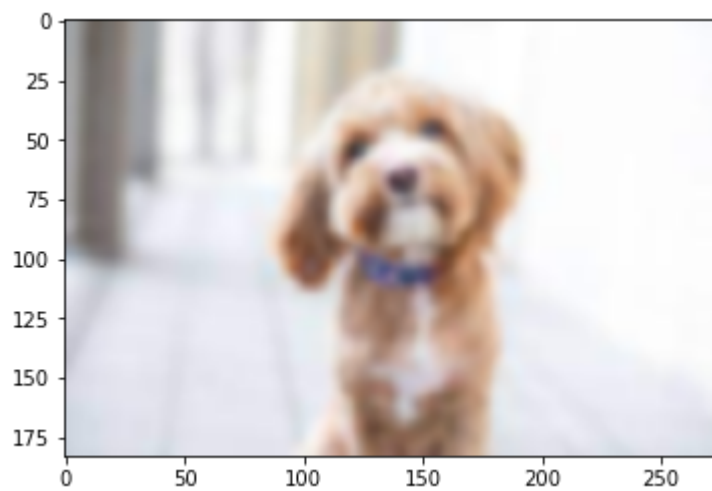
```
In [79]: # ImageFilter for using filter() function
from PIL import Image, ImageFilter

# Opening the image
# (R prefixed to string in order to deal with '\' in paths)
image = Image.open(r'C:/Users/hp/Downloads/a.jpg')

# Blurring image by sending the ImageFilter.
# GaussianBlur predefined kernel argument
image = image.filter(ImageFilter.GaussianBlur)

# Displaying the image
plt.imshow(image)
```

Out[79]: <matplotlib.image.AxesImage at 0x205eabca850>



```
In [81]: from PIL import Image, ImageFilter

image = Image.open(r'C:/Users/hp/Downloads\a.jpg')

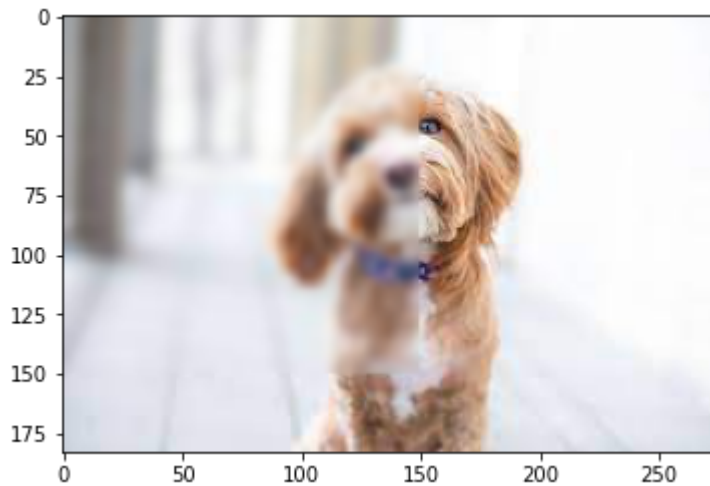
# Cropping the image
smol_image = image.crop((0, 0, 150, 150))

# Blurring on the cropped image
blurred_image = smol_image.filter(ImageFilter.GaussianBlur)

# Pasting the blurred image on the original image
image.paste(blurred_image, (0,0))

# Displaying the image
image.save('output.png')
plt.imshow(image)
```

Out[81]: <matplotlib.image.AxesImage at 0x205eac23f40>



In []:

In []:

In []: