In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as pim
```

In [2]:

```
# initializing image
img_path = "tokyo-japan.jpg"
image = pim.imread(img_path)
plt.imshow(image)
plt.show()
```

In [3]:

```
# discovering limits and parameters
height, width, n_colors = image.shape
print("height: {}\twidth: {}\tn_colors: {}".format(height, width, n_colors))
print("range de pixels: {}-{}".format(np.amin(image), np.amax(image)))
print(image.dtype)
# turn image from uint8 array to float array
image = image/255
print(image.dtype)
```

```
height: 1535 width: 2300 n_colors: 3
range de pixels: 0-255
uint8
float64
```

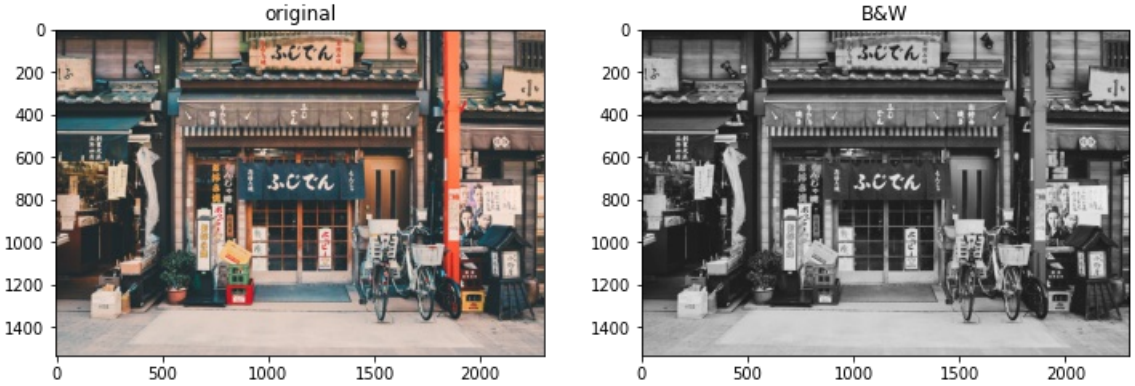
In [4]:

```
print("Altering image pixels...")
```

```
Altering image pixels...
```

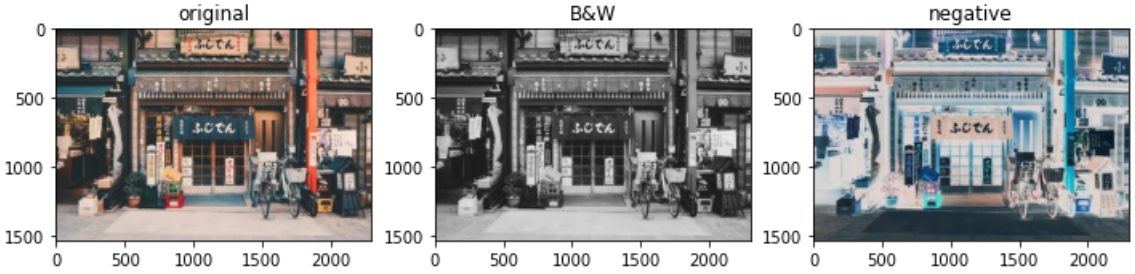
In [5]:

```
# Changing image to black and white, altering it's luminosity values
def to_gray(pixels):
    filter = [0.2126, 0.7152, 0.0722]
    return np.dot(pixels, filter)

gray_image = to_gray(image)
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
axes[0].set_title("original")
axes[0].imshow(image)
axes[1].set_title("B&W")
axes[1].imshow(gray_image, cmap="gray")
plt.show()
```



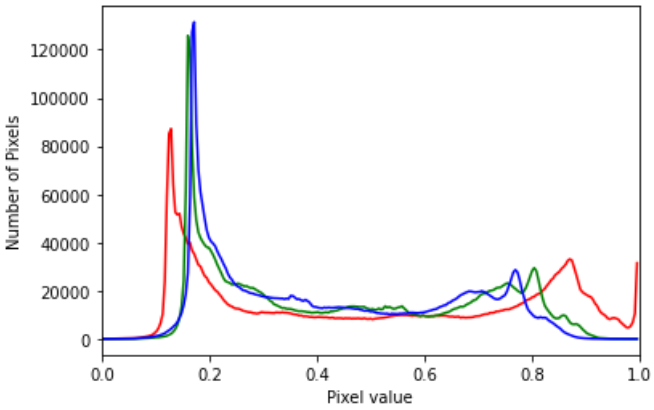
In [6]:

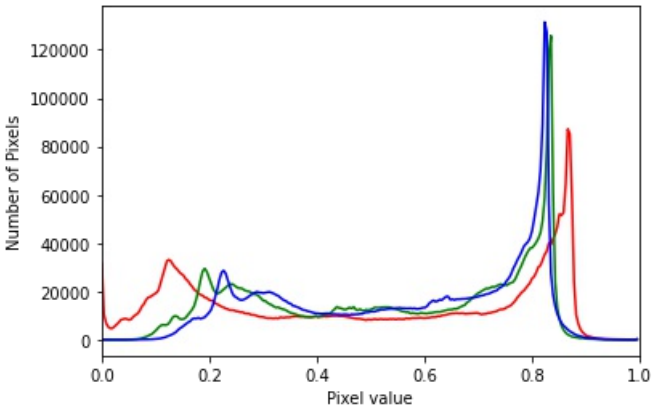
```
# Let's turn the original image to negative
# Since we normalized the original rgb pixel values for the array, all we need to do to shift it's color
# is subtract all of them by one, getting the respective rgb values of the complement of the original pi
# would turn that pixel to white (255)
negative = 1 - image
fig, axes = plt.subplots(1, 3, figsize=(12, 6))
axes[0].set_title("original")
axes[0].imshow(image)
axes[1].set_title("B&W")
axes[1].imshow(gray_image, cmap="gray")
axes[2].set_title("negative")
axes[2].imshow(negative)
plt.show()
```



In [7]:

```
# Since we have both the original picture and the negative, it will be interesting to observe it's pixel
# They should be the exact oposite of each other

# tuple to select colors of each channel line
colors = ("red", "green", "blue")
channel_ids = (0, 1, 2)

# create the histogram plot, with three lines, one for each color
def color_histogram(image):
    plt.xlim([0, 1])
    for channel_id, c in zip(channel_ids, colors):
        histogram, bin_edges = np.histogram(
            image[:, :, channel_id], bins=256, range=(0, 1)
        )
        plt.plot(bin_edges[0:-1], histogram, color=c)

    plt.xlabel("Pixel value")
    plt.ylabel("Number of Pixels")

    plt.show()

color_histogram(image)
color_histogram(negative)
```


Number of Pixels
120000
100000
80000
60000
40000
20000
0
0.0
0.2
0.4
0.6
0.8
1.0
Pixel value


Number of Pixels
120000
100000
80000
60000
40000
20000
0
0.0
0.2
0.4
0.6
0.8
1.0
Pixel value

In [ ]: