```
import cv2
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
def display_img(img,cmap=None):
    fig = plt.figure(figsize=(12,10))
    ax = fig.add_subplot(111)
    ax.imshow(img,cmap)
```

TASK: Open and display the giaraffes.jpg image that is located in the DATA folder.

```
from PIL import Image
```

```
IMG = Image.open('/content/giraffes.jpg')
```

```
type(IMG)
```

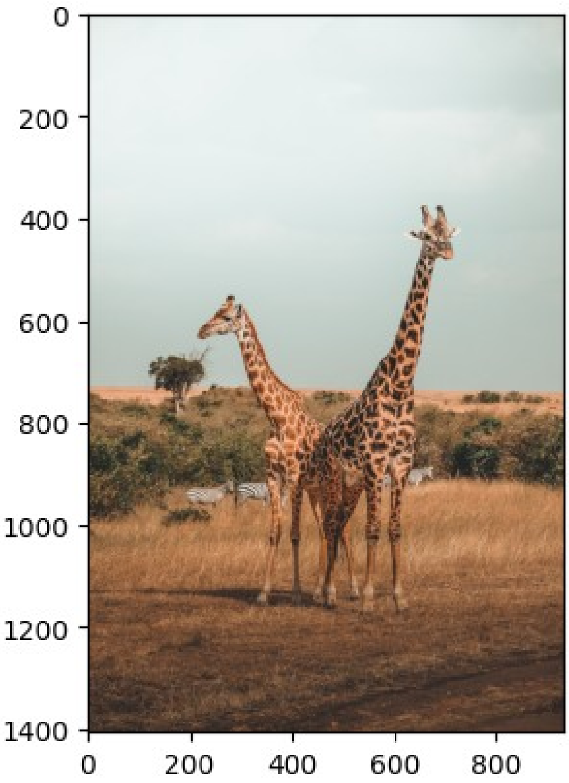
```
PIL.JpegImagePlugin.JpegImageFile
```

```
IMG_arr = np.asarray(IMG)
IMG_arr.shape
```

```
(1405, 933, 3)
```

```
plt.imshow(IMG_arr)
```

```
<matplotlib.image.AxesImage at 0x7f09b3453430>
```



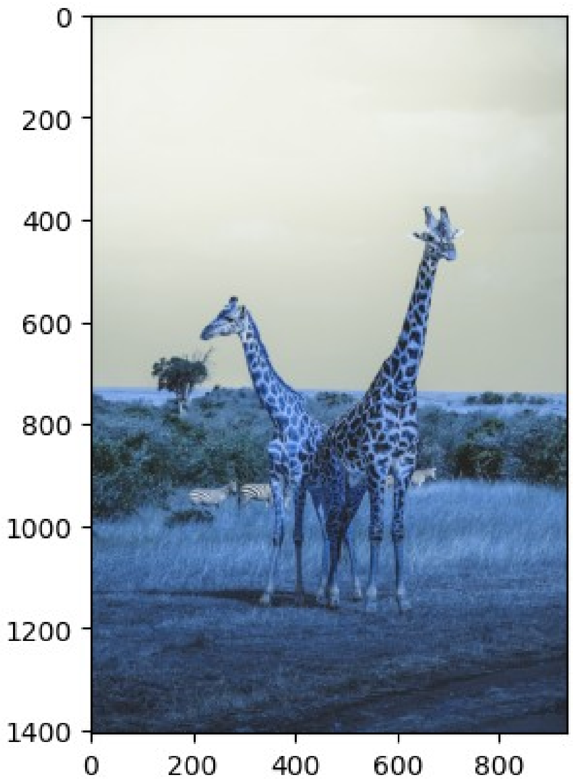
Apply a binary threshold onto the image.

```
import cv2
```

```
# Adding the 0 flag to read it in black and white
img = cv2.imread('/content/giraffes.jpg')
```

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

```
<matplotlib.image.AxesImage at 0x7f09b33c5670>
```



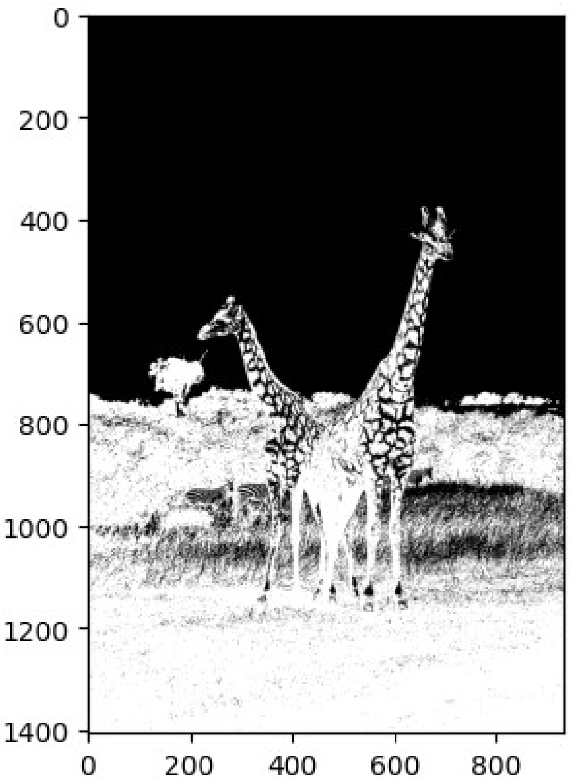
```
img = cv2.imread('/content/giraffes.jpg', cv2.IMREAD_GRAYSCALE)

# Apply binary threshold with a threshold value of 127
threshold_value = 127
_, binary_img = cv2.threshold(img, threshold_value, 255,
cv2.THRESH_BINARY)

# Save the resulting binary image
cv2.imwrite('/content/giraffes.jpg', binary_img)
```

```
True
```

```
plt.imshow(binary_img, cmap='binary')
plt.show()
```

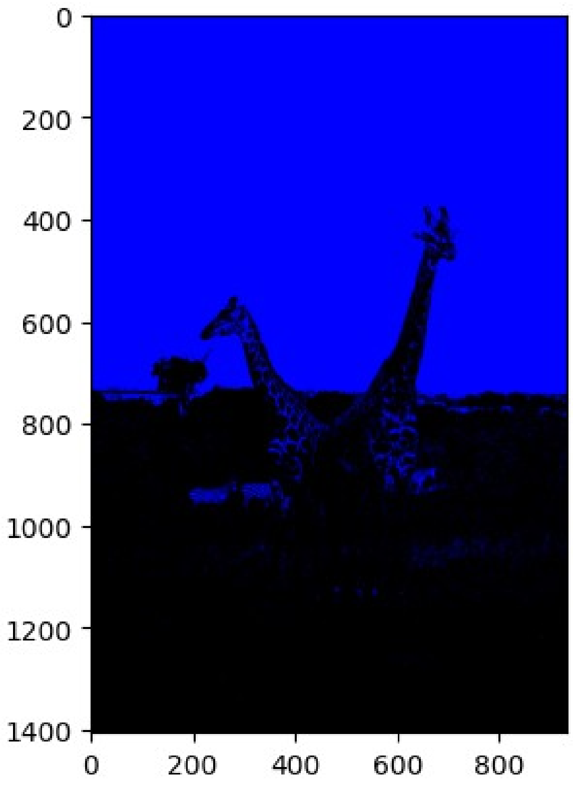


TASK: Open the giaraffes.jpg file from the DATA folder and convert its colorspace to HSV and display the image.

```
img = cv2.imread('/content/giraffes.jpg')

hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

# Plot the HSV image with manual intensity range
plt.imshow(hsv_img, vmin=0, vmax=255)
plt.show()
```



Create a low pass filter with a 4 by 4 Kernel filled with values of 1/10 (0.01) and then use 2-D Convolution to blur the giraffer image (displayed in normal RGB)

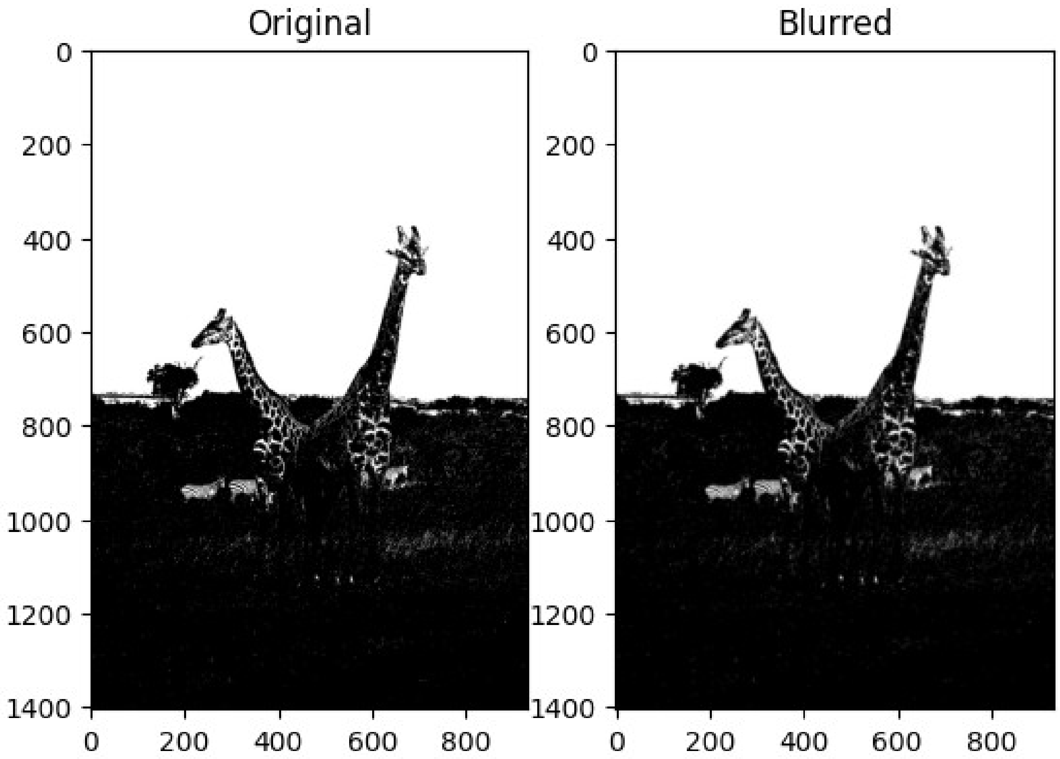
```
# Load the giraffe image in RGB format
img = cv2.imread('/content/giraffes.jpg', cv2.IMREAD_COLOR)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Define the 4x4 kernel for low pass filter
kernel_size = 4
kernel = np.ones((kernel_size, kernel_size), dtype=np.float32) /
(kernel_size * kernel_size)

# Apply the kernel using 2-D convolution
blurred_img = cv2.filter2D(img, -1, kernel)

# Display the original and blurred images
fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.imshow(img)
ax1.set_title('Original')
ax2.imshow(blurred_img)
ax2.set_title('Blurred')

plt.show()
```

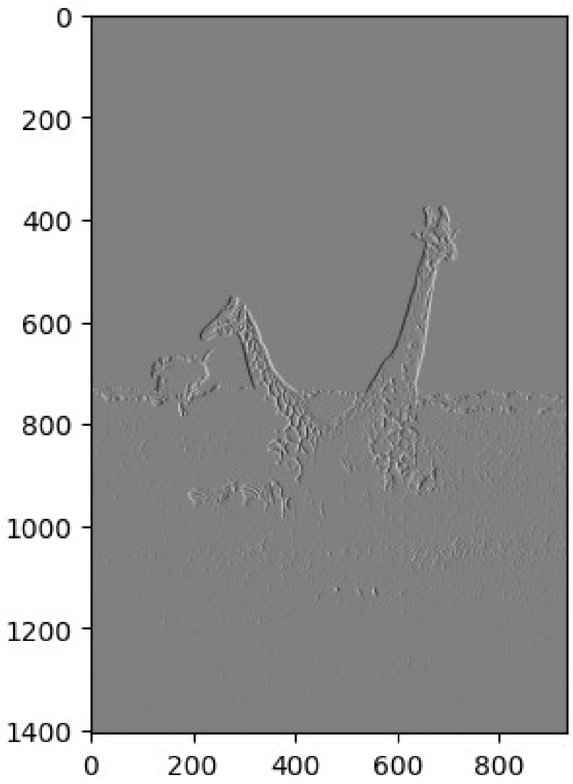


TASK: Create a Horizontal Sobel Filter (sobelx from our lecture) with a kernel size of 5 to the grayscale version of the giaraffes image and then display the resulting gradient filtered version of the image.

```
# Load the giraffe image in grayscale
img = cv2.imread('/content/giraffes.jpg', cv2.IMREAD_GRAYSCALE)

# Define the horizontal Sobel filter
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=5)

# Display the filtered image
plt.imshow(sobelx, cmap='gray')
plt.show()
```



TASK: Plot the color histograms for the RED, BLUE, and GREEN channel of the giaraffe image. Pay careful attention to the ordering of the channels.

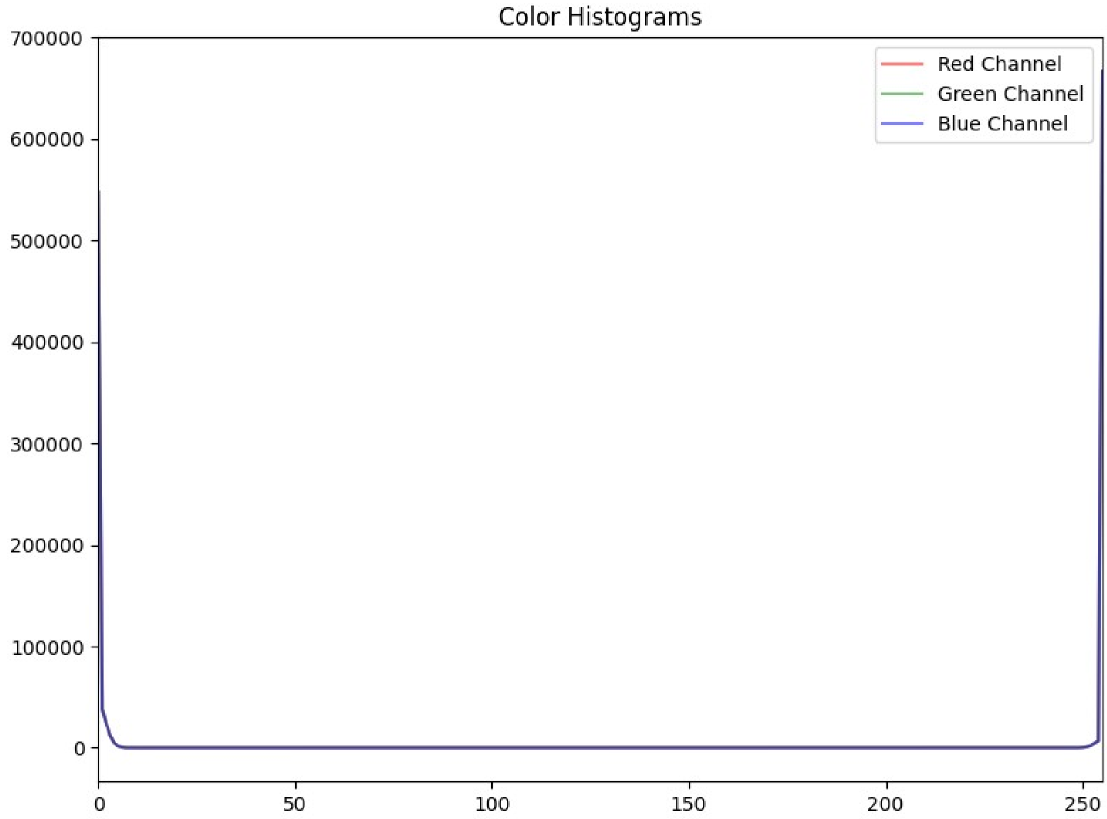
```
# Load the giraffe image
img = cv2.imread('/content/giraffes.jpg')

# Split the image into its color channels
b, g, r = cv2.split(img)

# Set the histogram parameters
hist_size = 256
hist_range = (0, 256)

# Compute the histograms for each color channel
r_hist = cv2.calcHist([r], [0], None, [hist_size], hist_range)
g_hist = cv2.calcHist([g], [0], None, [hist_size], hist_range)
b_hist = cv2.calcHist([b], [0], None, [hist_size], hist_range)

# Plot the color histograms
plt.figure(figsize=(8, 6))
plt.plot(r_hist, color='red', alpha=0.5, label='Red Channel')
plt.plot(g_hist, color='green', alpha=0.5, label='Green Channel')
plt.plot(b_hist, color='blue', alpha=0.5, label='Blue Channel')
plt.xlim([0, hist_size-1])
plt.legend()
plt.title('Color Histograms')
plt.tight_layout()
plt.show()
```


Color Histograms
Red Channel
Green Channel
Blue Channel