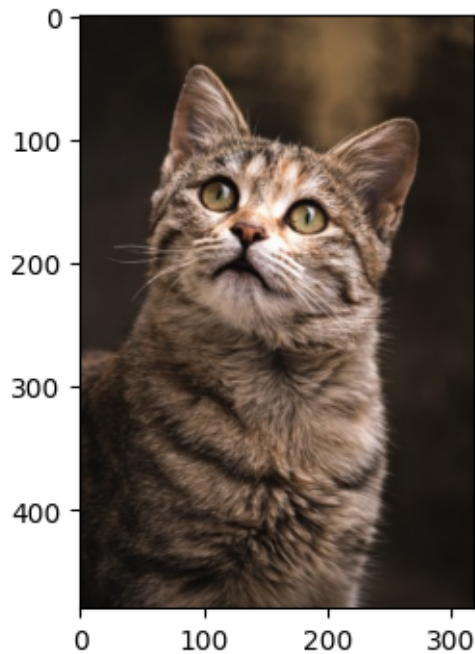


## openCV-Basic-2

March 27, 2023

```
[1]: import matplotlib.pyplot as plt
import numpy as np
import imutils
import cv2
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

```
[2]: plt.figure(figsize=(6, 4))
imagedata = plt.imread("CuteCat.jpg")
plt.imshow(imagedata)
plt.grid(False)
plt.show()
```



```
[3]: len(imagedata.shape)
```

```
[3]: 3
```

```
[4]: len(imagedata.shape)
```

```
[4]: 3
```

```
[5]: def catimageShow(imageTitle, image):  
    if len(image.shape) == 3:  
        imageVariable = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
    # if image.shape[2] == 1:  
    # imageVariable = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)  
    plt.figure(figsize=(4, 2))  
    plt.imshow(imageVariable)  
    plt.title(imageTitle)  
    plt.show()
```

```
[6]: imagedata.shape
```

```
[6]: (480, 320, 3)
```

```
[7]: cv2.split(imagedata)
```

```
[7]: (array([[26, 26, 26, ..., 73, 72, 71],  
          [26, 26, 26, ..., 73, 71, 70],  
          [25, 26, 26, ..., 71, 69, 68],  
          ...,  
          [15, 15, 16, ..., 29, 28, 28],  
          [14, 14, 14, ..., 29, 29, 29],  
          [14, 13, 13, ..., 30, 29, 29]], dtype=uint8),  
 array([[21, 21, 21, ..., 56, 55, 54],  
          [21, 21, 21, ..., 56, 54, 53],  
          [20, 21, 21, ..., 55, 53, 52],  
          ...,  
          [13, 13, 14, ..., 24, 23, 23],  
          [14, 14, 14, ..., 24, 24, 24],  
          [14, 13, 13, ..., 25, 24, 24]], dtype=uint8),  
 array([[18, 18, 18, ..., 40, 39, 38],  
          [18, 18, 18, ..., 40, 38, 37],  
          [17, 18, 18, ..., 39, 37, 36],  
          ...,  
          [16, 16, 15, ..., 20, 19, 19],  
          [16, 16, 16, ..., 20, 20, 20],  
          [16, 15, 15, ..., 21, 20, 20]], dtype=uint8))
```

```
[8]: def imageHistogram(image, imageTitle, mask = None):  
    color_Channel = cv2.split(image)  
    color_com = ('b', 'g', 'r')  
    plt.figure(figsize = (4, 2))
```

```

plt.title(imageTitle)
plt.xlabel("Bins")
plt.ylabel("Pixel Information")

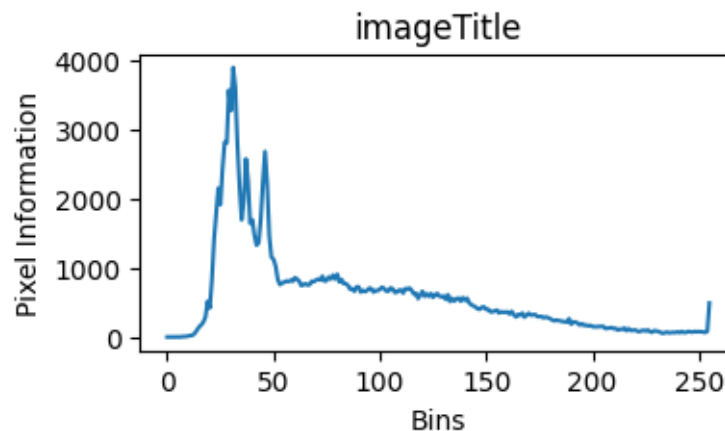
for (color_Channel, color_com) in zip(color_Channel, color_com):
    histogram = cv2.calcHist([color_Channel], [0], mask, [256], [0, 256])
    plt.plot(histogram, color = color_com)
    plt.xlim([0, 256])
    plt.show()

```

```

[9]: image_argument = {"Image": "CuteCat.jpg"}
image = cv2.imread(image_argument["Image"])
image= cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
smamplehisto = cv2.calcHist([image],[0], None, [256], [0, 256])
plt.figure(figsize = (4, 2))
plt.title("imageTitle")
plt.xlabel("Bins")
plt.ylabel("Pixel Information")
plt.plot(smamplehisto)
plt.show()

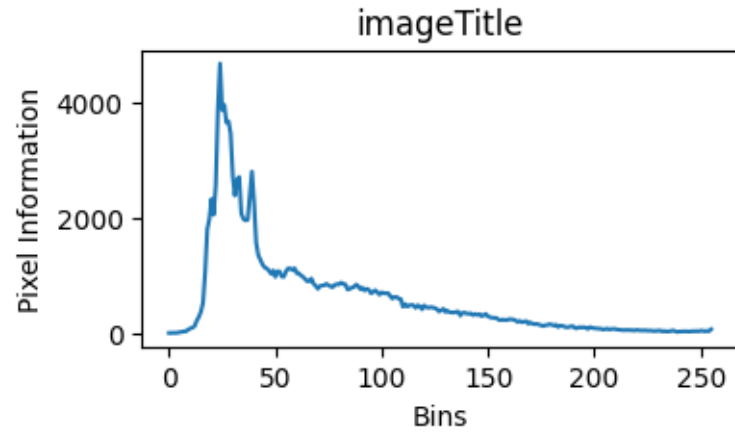
```



```

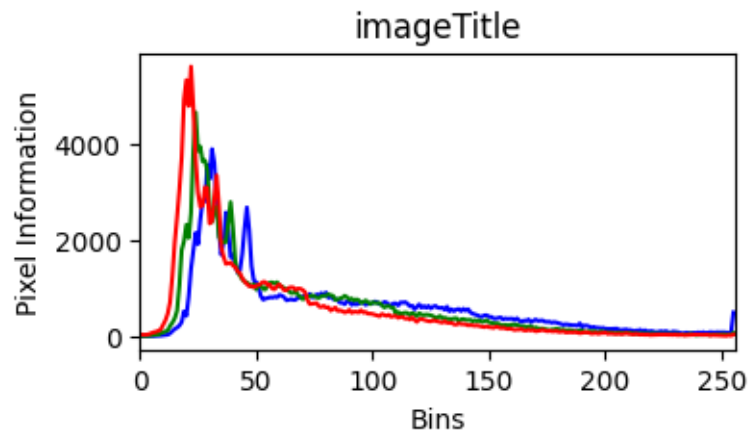
[10]: image_argument = {"Image": "CuteCat.jpg"}
image = cv2.imread(image_argument["Image"])
image= cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
smamplehisto = cv2.calcHist([image],[1], None, [256], [0, 256])
plt.figure(figsize = (4, 2))
plt.title("imageTitle")
plt.xlabel("Bins")
plt.ylabel("Pixel Information")
plt.plot(smamplehisto) #line
plt.show()

```



```
[11]: color_Channel = cv2.split(image)
color_com = ('b', 'g', 'r')
plt.figure(figsize = (4, 2))
plt.title("imageTitle")
plt.xlabel("Bins")
plt.ylabel("Pixel Information")

for (color_Channel, color_com) in zip(color_Channel, color_com):
    histogram = cv2.calcHist([color_Channel], [0],None, [256], [0, 256])
    plt.plot(histogram, color = color_com)
    plt.xlim([0, 256])
    #plt.show()
```

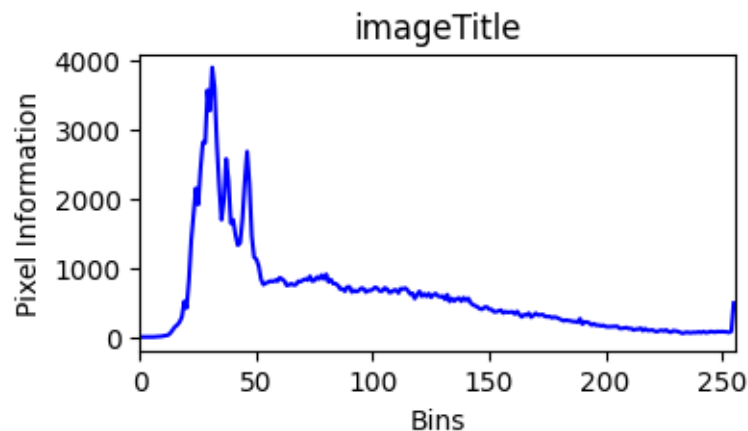


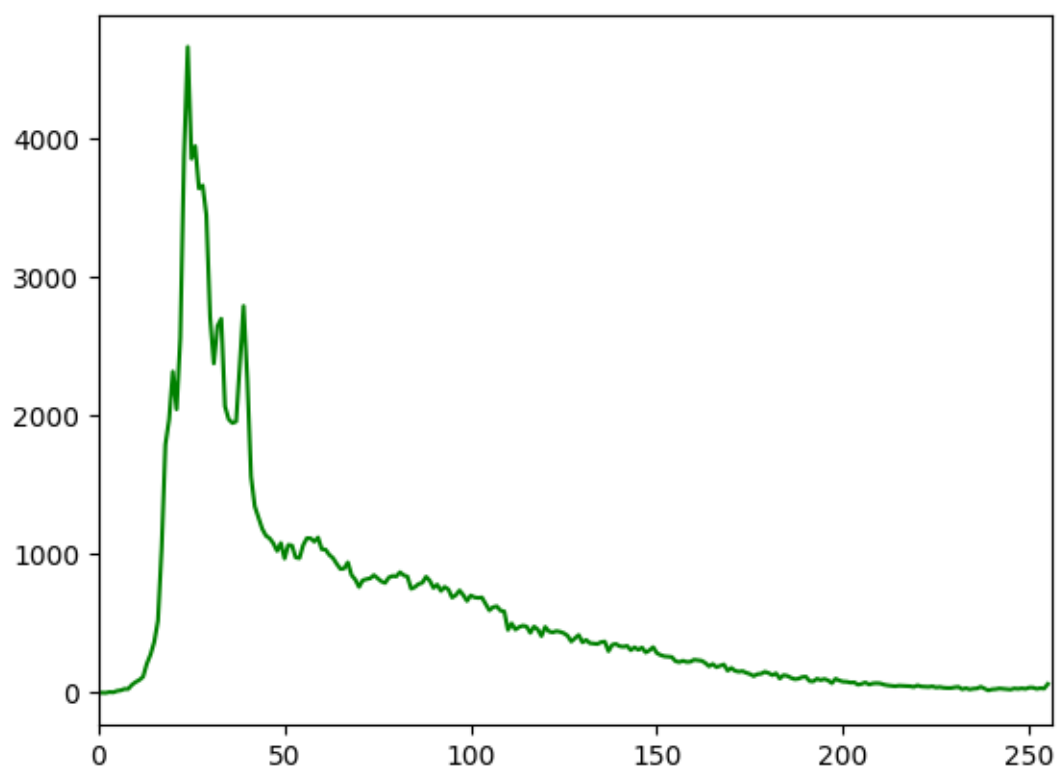
```
[12]: cv2.split(image)
```

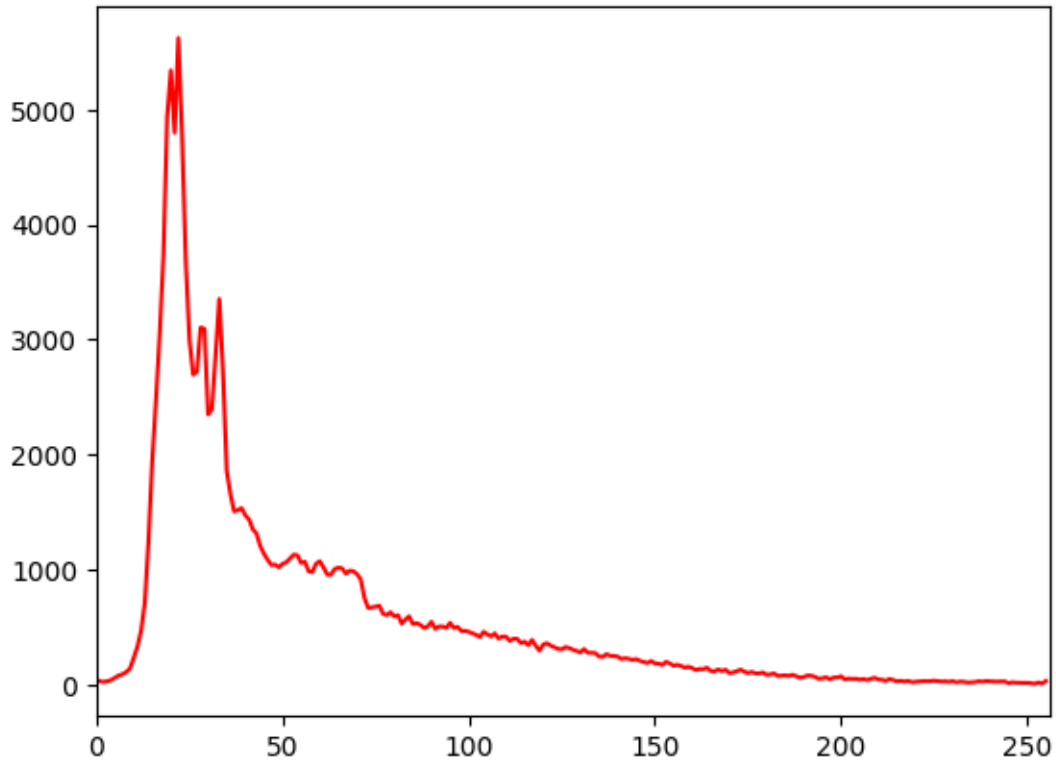
```
[12]: (array([[26, 26, 26, ..., 73, 72, 71],
             [26, 26, 26, ..., 73, 71, 70],
             [25, 26, 26, ..., 71, 69, 68],
             ...,
             [15, 15, 16, ..., 29, 28, 28],
             [14, 14, 14, ..., 29, 29, 29],
             [14, 13, 13, ..., 30, 29, 29]]), dtype=uint8),
      array([[21, 21, 21, ..., 56, 55, 54],
             [21, 21, 21, ..., 56, 54, 53],
             [20, 21, 21, ..., 55, 53, 52],
             ...,
             [13, 13, 14, ..., 24, 23, 23],
             [14, 14, 14, ..., 24, 24, 24],
             [14, 13, 13, ..., 25, 24, 24]]), dtype=uint8),
      array([[18, 18, 18, ..., 40, 39, 38],
             [18, 18, 18, ..., 40, 38, 37],
             [17, 18, 18, ..., 39, 37, 36],
             ...,
             [16, 16, 15, ..., 20, 19, 19],
             [16, 16, 16, ..., 20, 20, 20],
             [16, 15, 15, ..., 21, 20, 20]]), dtype=uint8))
```

```
[13]: color_Channel = cv2.split(image)
      color_com = ('b', 'g', 'r')
      plt.figure(figsize = (4, 2))
      plt.title("imageTitle")
      plt.xlabel("Bins")
      plt.ylabel("Pixel Information")

      for (color_Channel, color_com) in zip(color_Channel, color_com):
          histogram = cv2.calcHist([color_Channel], [0],None, [256], [0, 256])
          plt.plot(histogram, color = color_com)
          plt.xlim([0, 256])
          plt.show()
```

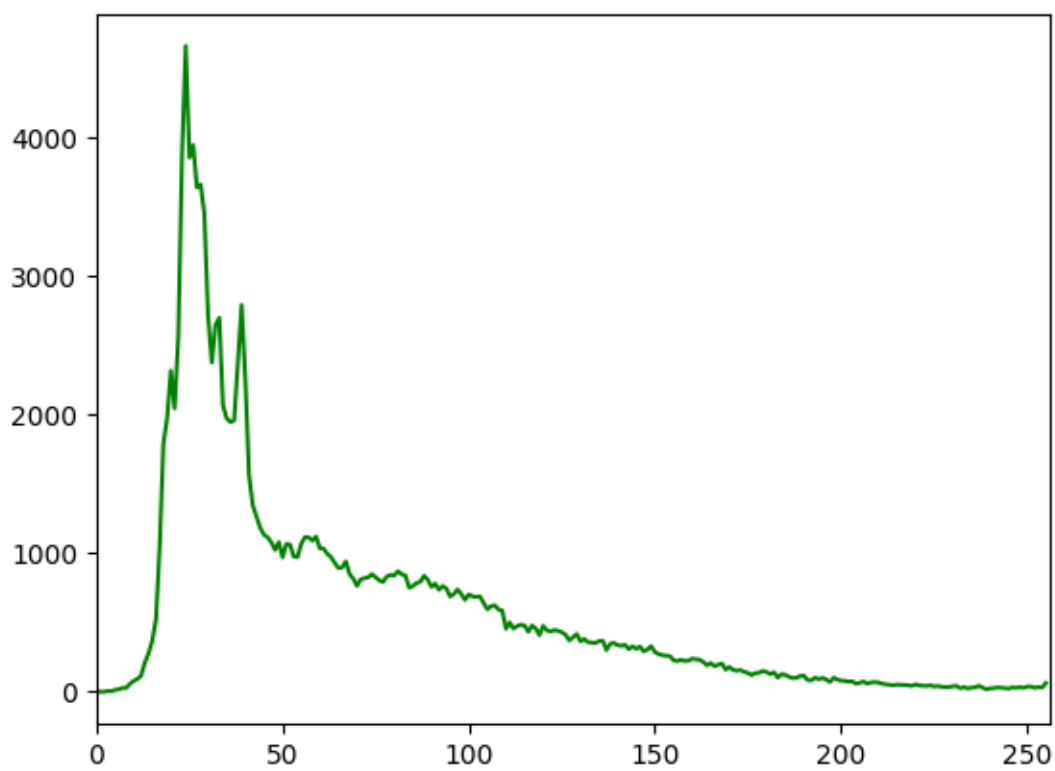
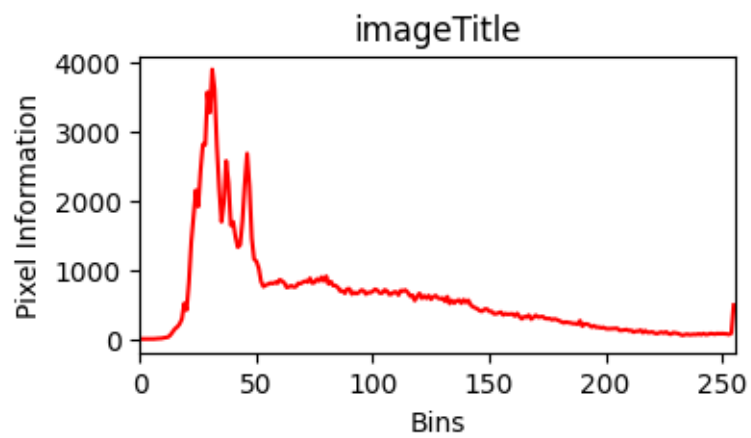




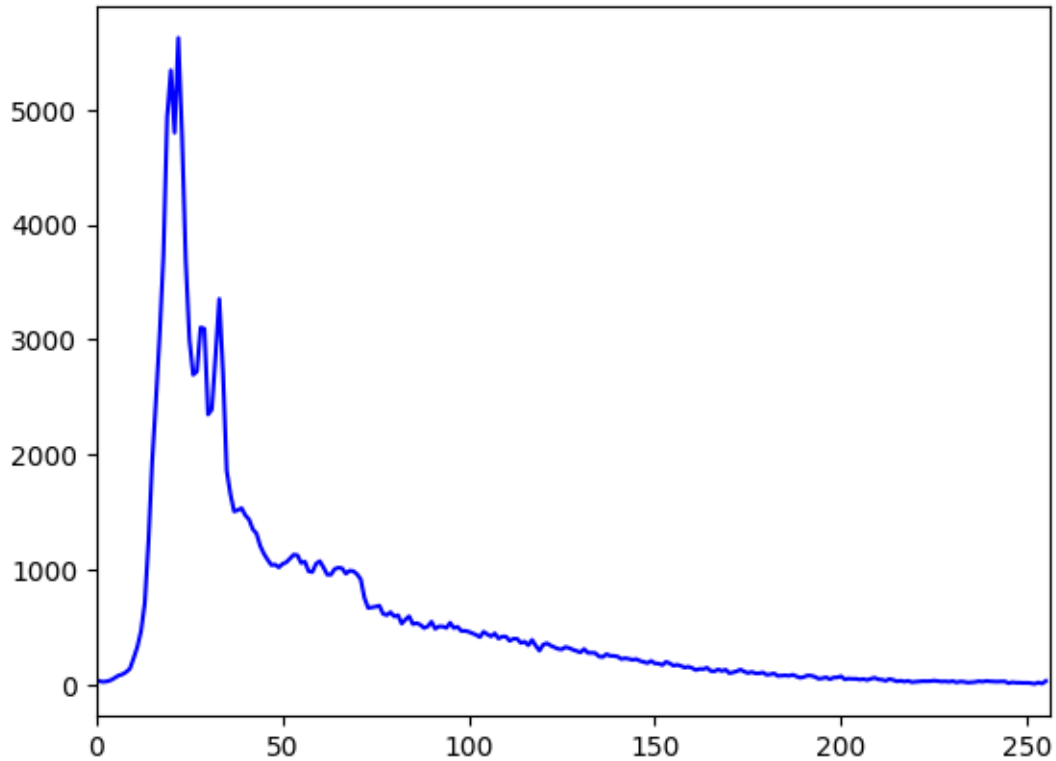


```
[14]: color_Channel = cv2.split(image)
color_com = ('r', 'g', 'b')
plt.figure(figsize = (4, 2))
plt.title("imageTitle")
plt.xlabel("Bins")
plt.ylabel("Pixel Information")

for (color_Channel, color_com) in zip(color_Channel, color_com):
    histogram = cv2.calcHist([color_Channel], [0],None, [256], [0, 256])
    plt.plot(histogram, color = color_com)
    plt.xlim([0, 256])
    plt.show()
```







```
[15]: color = ["B", "G", "R"]
fig = plt.figure(figsize=(20, 18))
ax = fig.add_subplot(131)
hist = cv2.calcHist([color_Channel[1], color_Channel[0]], [0, 1], None, [32, 32], [0, 256, 0, 256])
p = ax.imshow(hist, interpolation = "nearest")
ax.set_title("Image Generate for Green and Blue")
plt.colorbar(p)

ax = fig.add_subplot(132)
hist = cv2.calcHist([color_Channel[1], color_Channel[2]], [0, 1], None, [32, 32], [0, 256, 0, 256])
p = ax.imshow(hist, interpolation = "nearest")
ax.set_title("Image Generate for Green and Red")
plt.colorbar(p)

ax = fig.add_subplot(133)
hist = cv2.calcHist([color_Channel[0], color_Channel[2]], [0, 1], None, [32, 32], [0, 256, 0, 256])
```

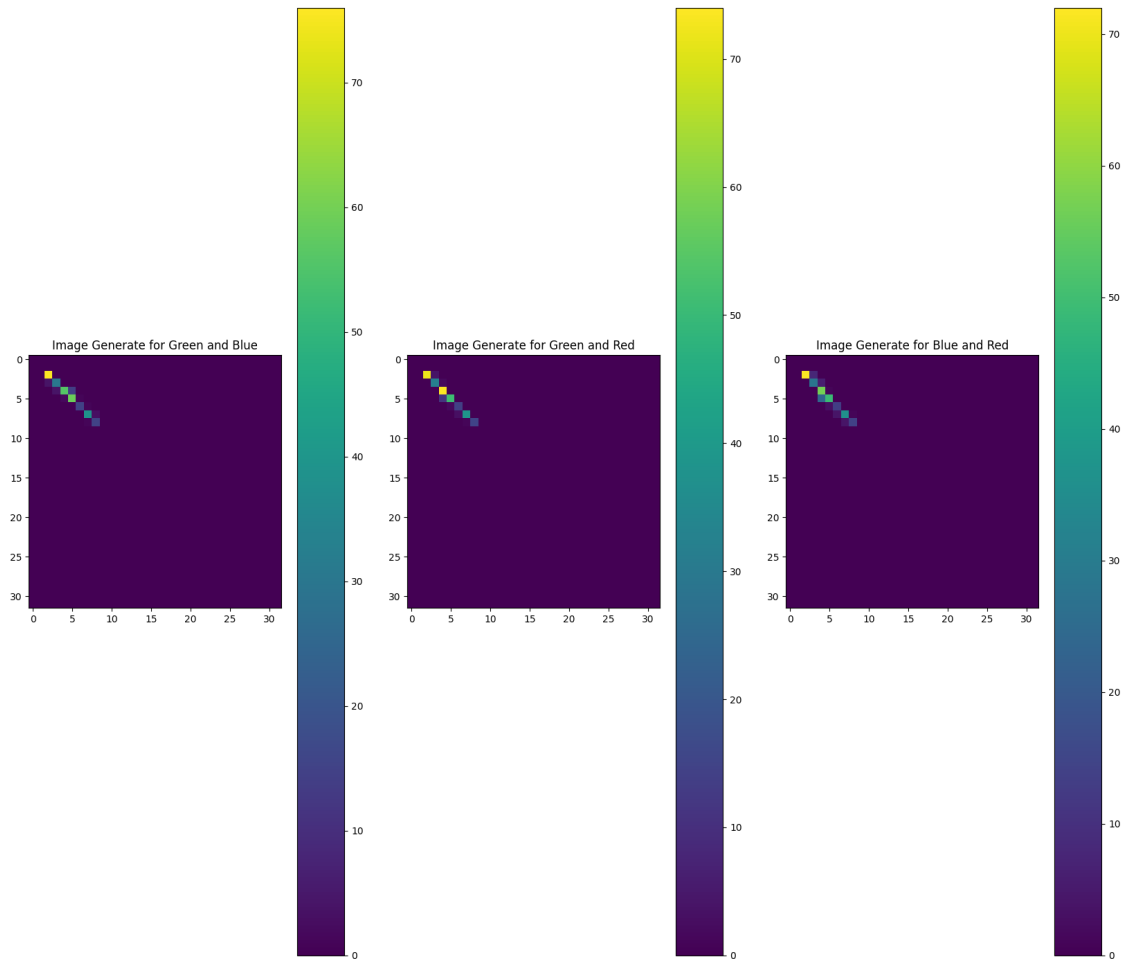
```

p = ax.imshow(hist, interpolation = "nearest")
ax.set_title("Image Generate for Blue and Red")
plt.colorbar(p)

print("Image Shape: {} with {}".format(hist.shape, hist.flatten().shape[0]))

```

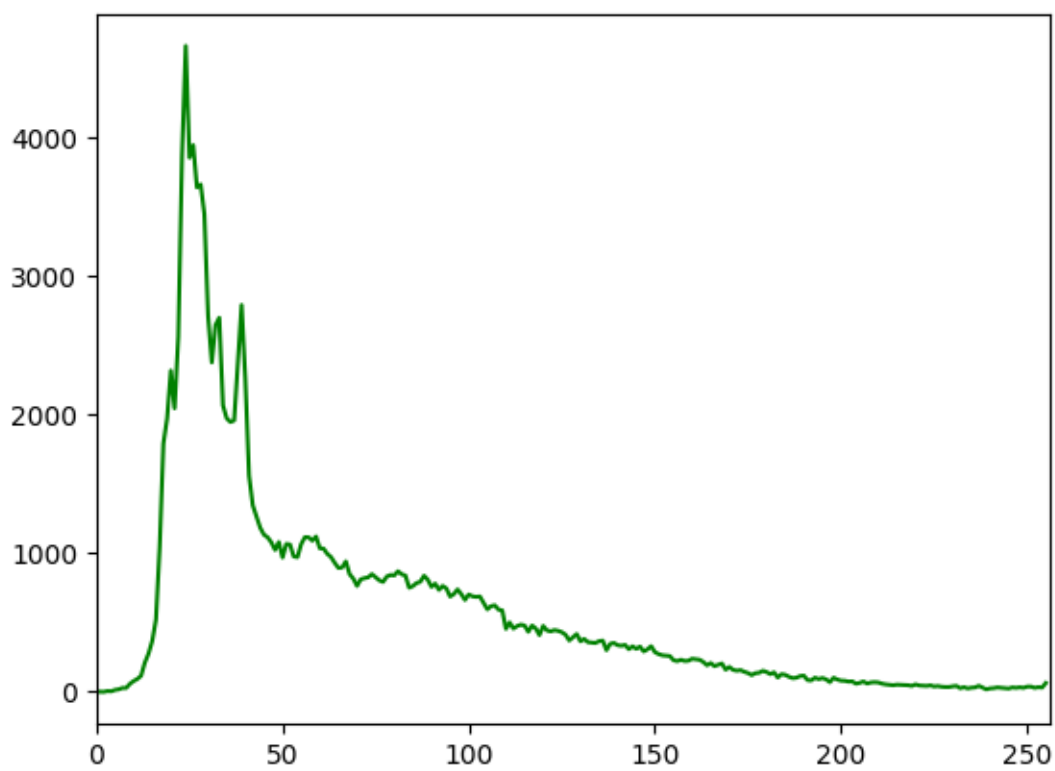
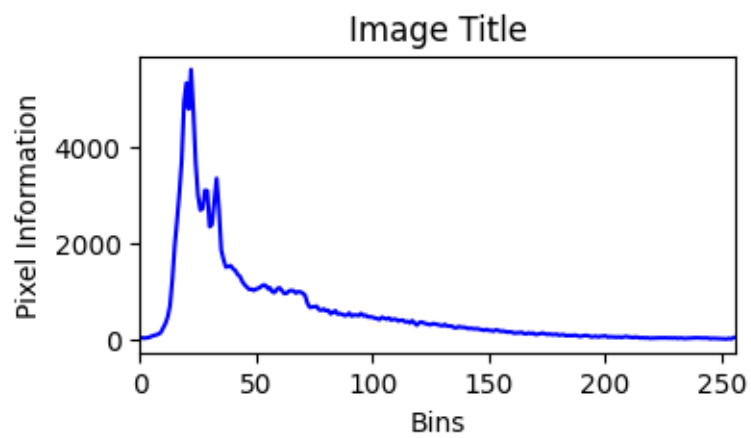
Image Shape: (32, 32) with 1024

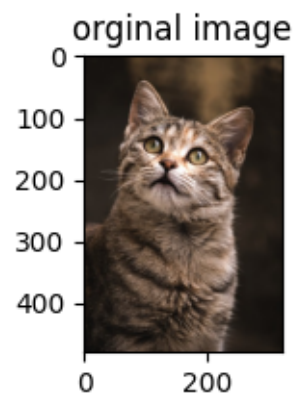
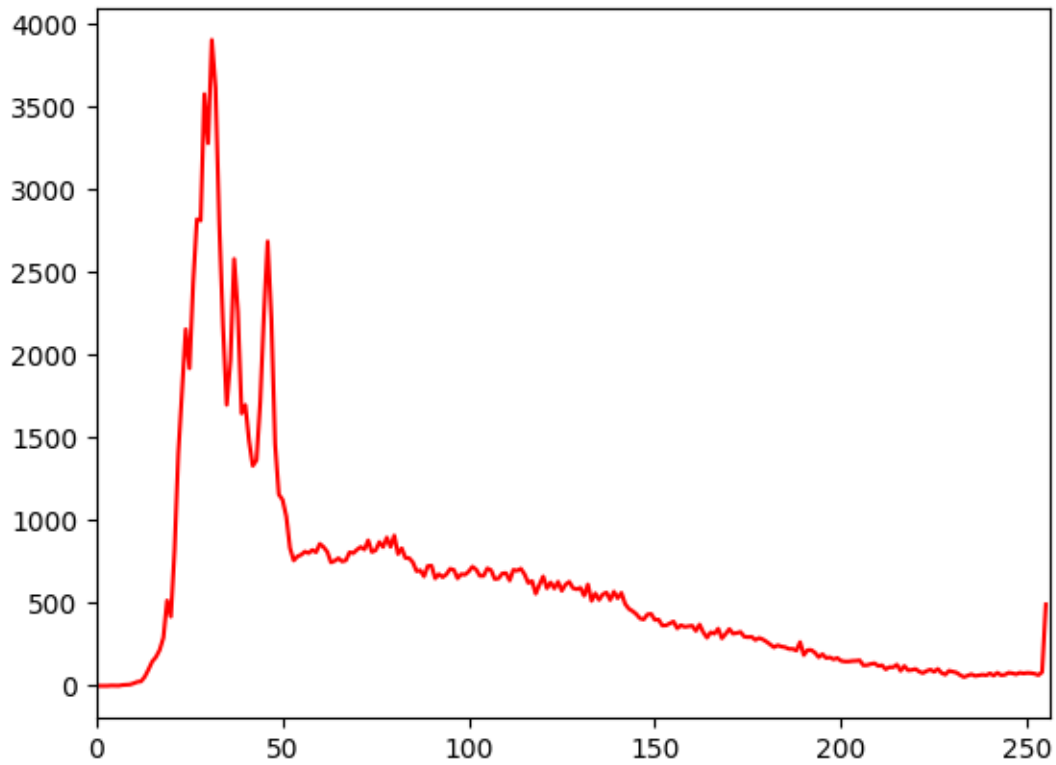


```

[16]: image = cv2.imread("CuteCat.jpg")
      imageHistogram(image, "Image Title")
      catimageShow("original image", image)

```





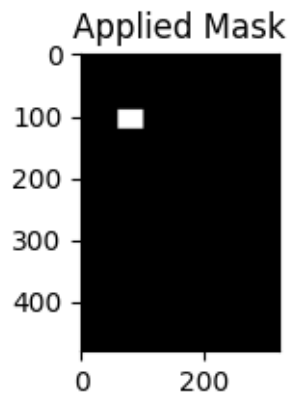
```
[17]: image.shape[:2]
```

```
[17]: (480, 320)
```

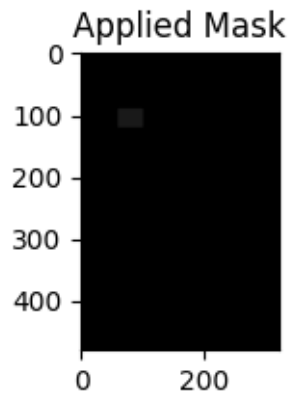
```
[18]: def catimageShow(imageTitle, image):  
      imageVariable = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
      plt.figure(figsize=(4, 2))
```

```
plt.imshow(imageVariable)
plt.title(imageTitle)
plt.show()
```

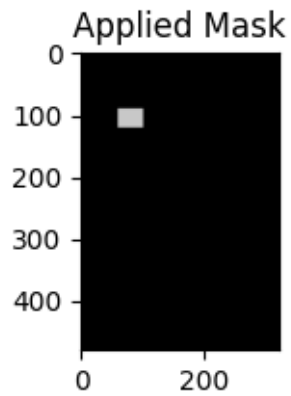
```
[19]: mask = np.zeros(image.shape[:2], dtype= "uint8")
cv2.rectangle(mask, (60, 90), (100, 120), 255, -1)
catimageShow("Applied Mask", mask)
```



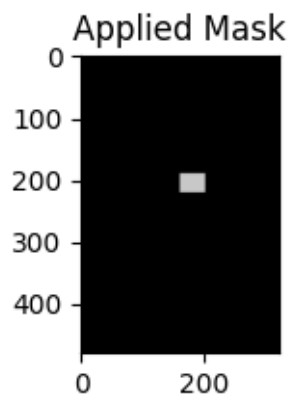
```
[20]: mask = np.zeros(image.shape[:2], dtype= "uint8")
cv2.rectangle(mask, (60, 90), (100, 120), 25, -1)
catimageShow("Applied Mask", mask)
```



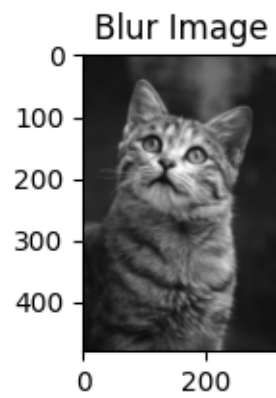
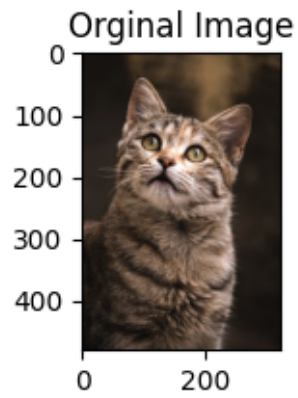
```
[21]: mask = np.zeros(image.shape[:2], dtype= "uint8")
cv2.rectangle(mask, (60, 90), (100, 120), 200, -1)
catimageShow("Applied Mask", mask)
```



```
[22]: mask = np.zeros(image.shape[:2], dtype= "uint8")  
      cv2.rectangle(mask, (160, 190), (200, 220), 200, -1)  
      catimageShow("Applied Mask", mask)
```



```
[23]: image = cv2.imread("CuteCat.jpg")  
      gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
      blurImage = cv2.GaussianBlur(gray, (5, 5), 0)  
      catimageShow("Original Image", image)  
      catimageShow("Blur Image", blurImage)
```



```
[24]: cannyBlurWide = cv2.Canny(blurImage, 20, 250)
      cannyBlurMiddle = cv2.Canny(blurImage, 15, 120)
      cannyBlurFit = cv2.Canny(blurImage, 240, 250)

      catimageShow("cannyBlurWide",cannyBlurWide)
      catimageShow("cannyBlurMiddle", cannyBlurMiddle)
      catimageShow("cannyBlurFit", cannyBlurFit)
```

