

cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])

images : it is the source image of type uint8 or float32 represented as “[img]”. **channels** : it is the index of channel for which we calculate histogram. For grayscale image, its value is [0] and color image, you can pass [0], [1] or [2] to calculate histogram of blue, green or red channel respectively. **mask** : mask image. To find histogram of full image, it is given as “None”. **histSize** : this represents our BIN count. For full scale, we pass [256]. **ranges** : this is our RANGE. Normally, it is [0,256].

In [19]:

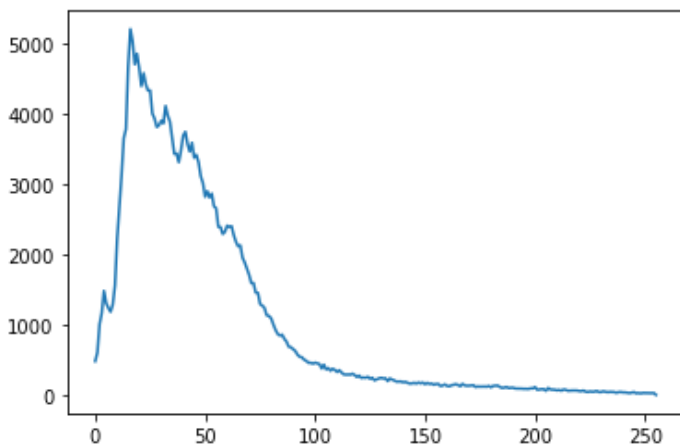
```
# importing required libraries of opencv
import cv2

# importing library for plotting
from matplotlib import pyplot as plt

# reads an input image
img = cv2.imread('parrots.jpg',0)

# find frequency of pixels in range 0-255
histr = cv2.calcHist([img],[0],None,[256],[0,255])

# show the plotting graph of an image
plt.plot(histr)
plt.show()
```



In [20]:

```
import cv2
from matplotlib import pyplot as plt
img = cv2.imread('parrots.jpg',2)

# alternative way to find histogram of an image
plt.hist(img.ravel(),256,[0,255])
plt.show()
```

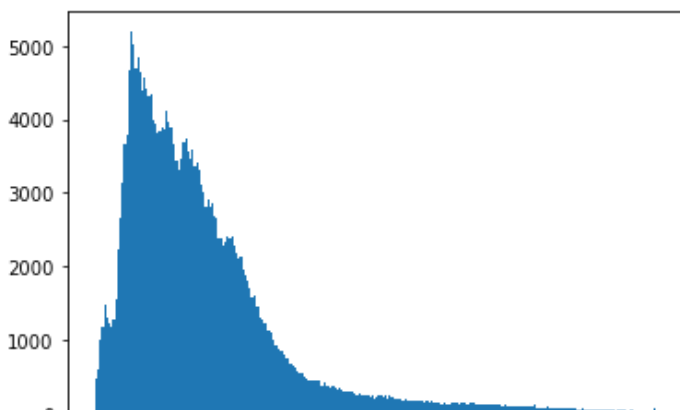


Image Pyramid

In [23]:

```
import cv2
import matplotlib.pyplot as plt

img = cv2.imread("parrots.jpg")

layer = img.copy()

for i in range(4):
    plt.subplot(2, 2, i + 1)

    # using pyrDown() function
    layer = cv2.pyrDown(layer)

plt.imshow(layer)
cv2.imshow("str(i)", layer)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

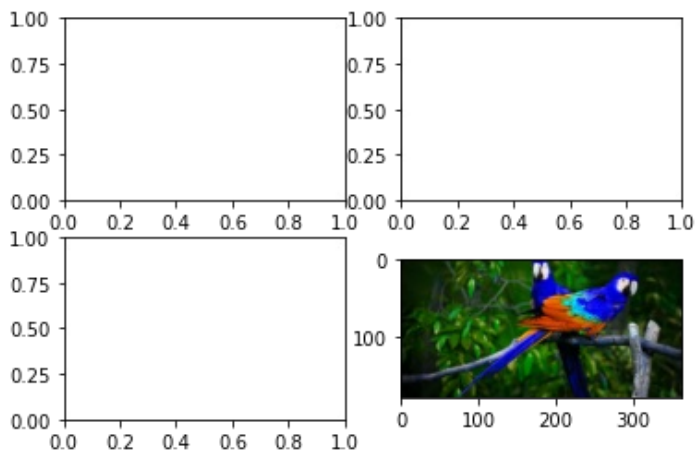


Image Translation

In [24]:

```
import cv2
import numpy as np

image = cv2.imread('moralis.jpg')

# Store height and width of the image
height, width = image.shape[:2]

quarter_height, quarter_width = height / 4, width / 4

T = np.float32([[1, 0, quarter_width], [0, 1, quarter_height]])

# We use warpAffine to transform
# the image using the matrix, T
img_translation = cv2.warpAffine(image, T, (width, height))

cv2.imshow("Originalimage", image)
cv2.imshow('Translation', img_translation)
cv2.waitKey()

cv2.destroyAllWindows()
```

