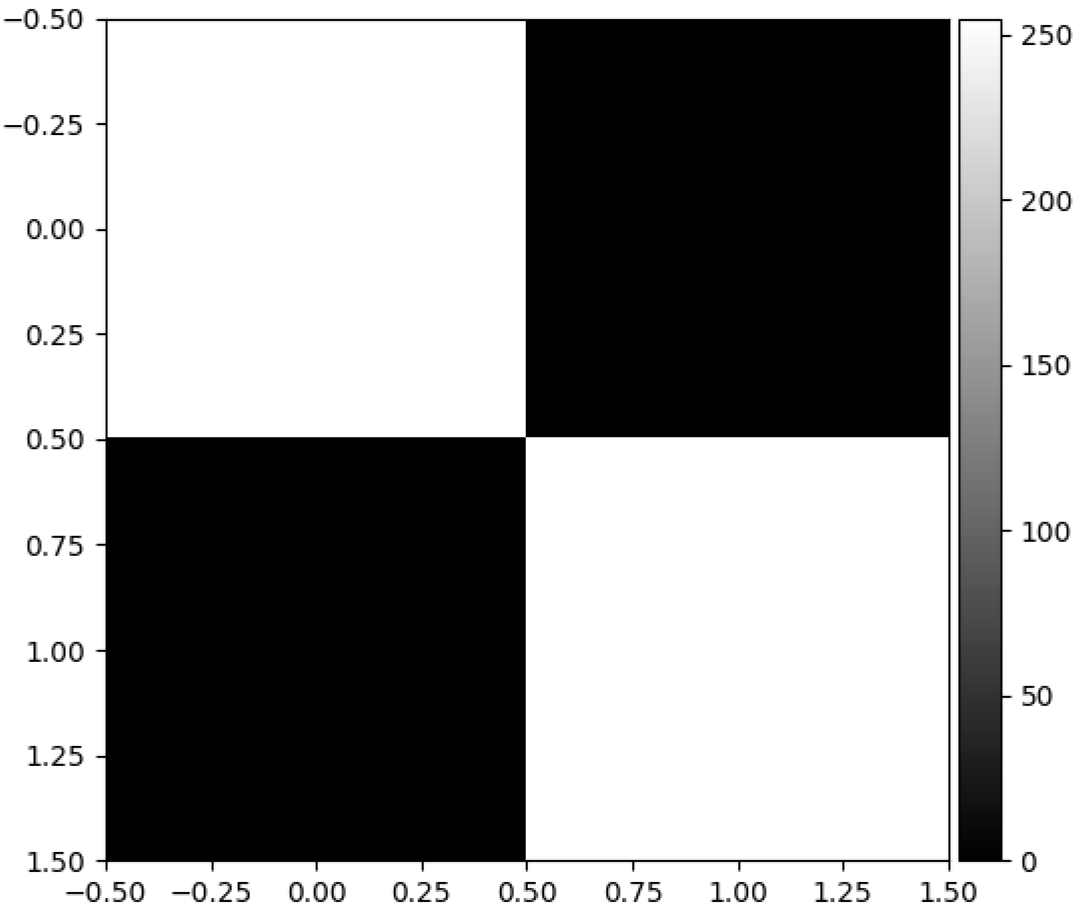
In [6]:

```
# First import the required python libraries
import cv2
import numpy as np
from PIL import Image
from skimage.io import imshow
import matplotlib.pyplot as plt
from skimage import img_as_uint
from skimage.io import imread
from skimage.color import rgb2hsv, rgb2gray

# 2x2 matrix image
array_1 = np.array([[255,0],[0,255]])
imshow(array_1, cmap='gray')
plt.show()
```

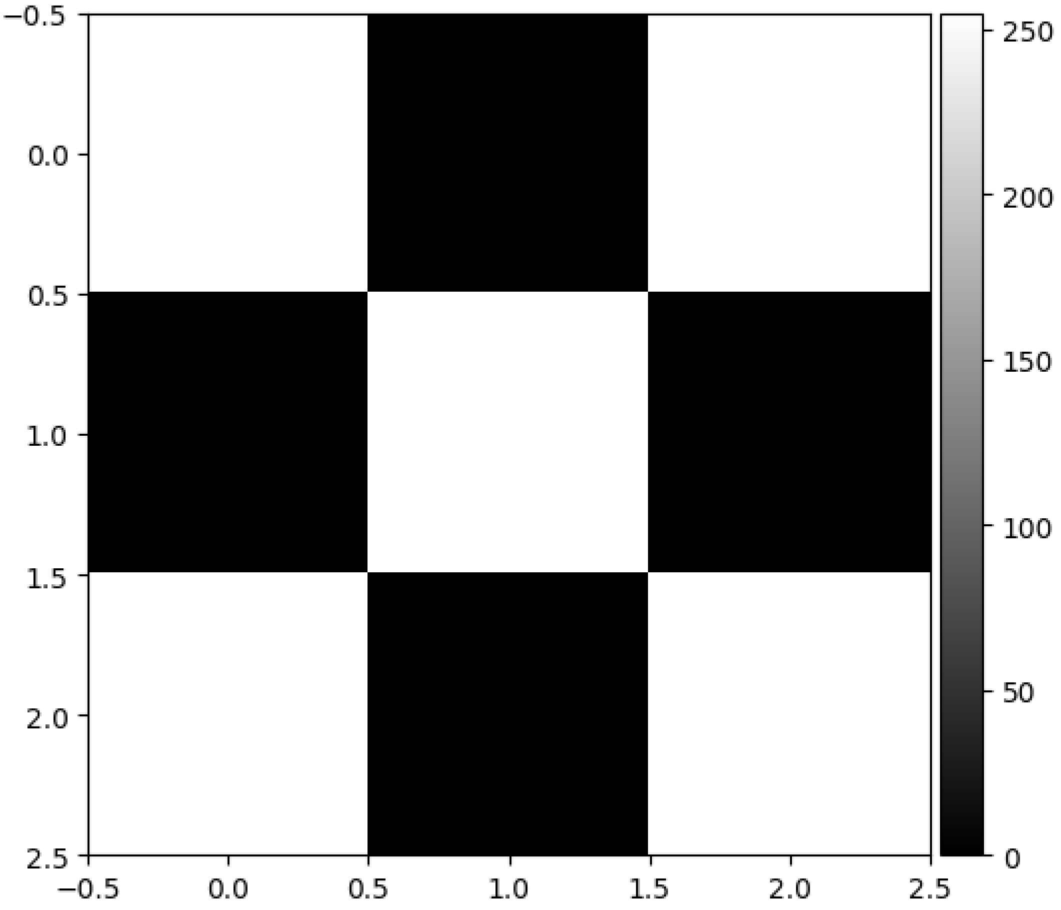


In [ ]:

In [ ]:

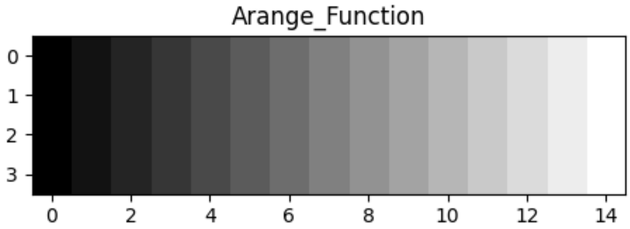
In [7]:

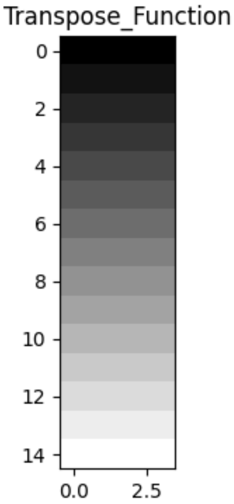
```
# 3x3 matrix image
array_2 = np.array([[255,0,255],[0,255,0],[255,0,255],])
imshow(array_2, cmap='gray')
plt.show()
```



In [8]:

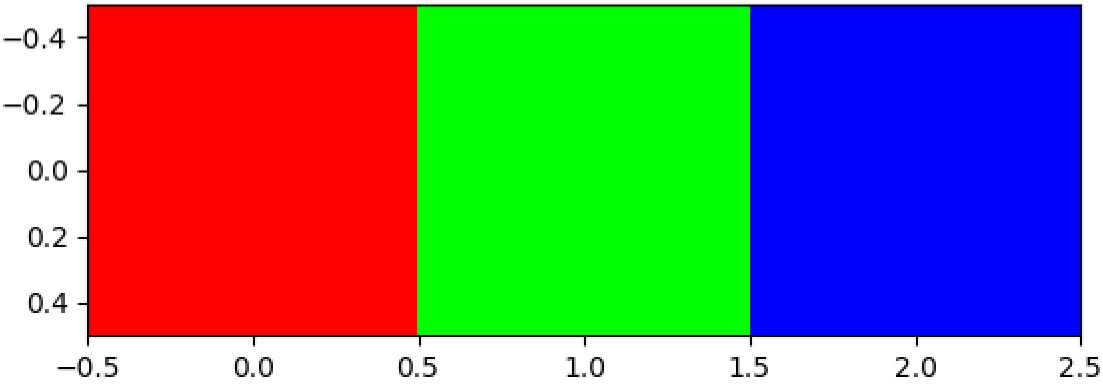
```
# normal image arrange matrix
array_spectrum = np.array([np.arange(0, 255, 17),
                           np.arange(0, 255, 17),
                           np.arange(0, 255, 17),
                           np.arange(0, 255, 17)])

fig, ax = plt.subplots(1, 2, figsize=(12, 4))

# arrange function
ax[0].imshow(array_spectrum, cmap='gray')
ax[0].set_title('Arange_Function')

# transpose function
ax[1].imshow(array_spectrum.T, cmap='gray')
ax[1].set_title('Transpose_Function')

plt.show()
```





In [9]:

```
# RGB image matrix
array_colors=np.array([[[255,0,0],[0,255,0],[0,0,255]]])
plt.imshow(array_colors)
plt.show()
```



In [10]:

```
# display the image
puppy = imread(r"C:\Users\navin\Downloads\puppy.jpg")
imshow(puppy)
```

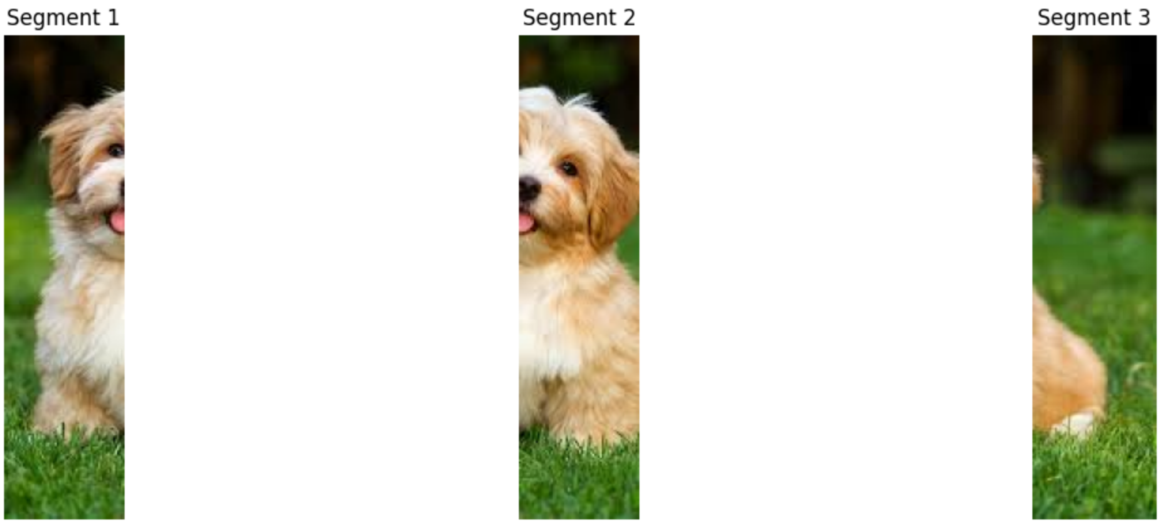
Out[10]:

```
<matplotlib.image.AxesImage at 0x2743c3f80b0>
```

In [16]:
```
puppy.shape
```

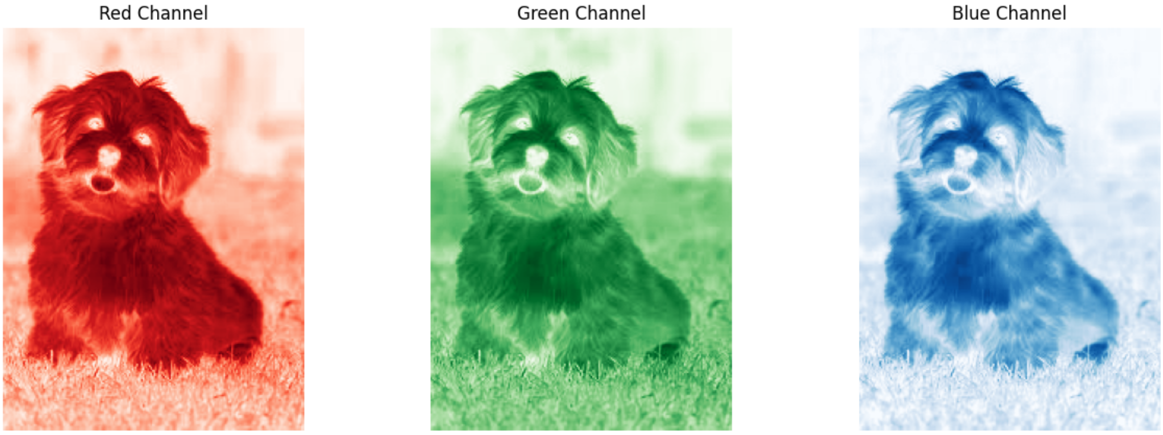
Out[16]:
```
(259, 194, 3)
```

In [21]:
```
# Open the puppy image
puppy_image = Image.open(r"C:\Users\navin\Downloads\puppy.jpg")

# Get the dimensions of the image
width, height = puppy_image.size

# Calculate the width for each segment
segment_width = width // 3

# Define the regions for the three segments
segment1 = (0, 0, segment_width, height)
segment2 = (segment_width, 0, 2 * segment_width, height)
segment3 = (2 * segment_width, 0, width, height)

# Crop the image into three segments
puppy_segment1 = puppy_image.crop(segment1)
puppy_segment2 = puppy_image.crop(segment2)
puppy_segment3 = puppy_image.crop(segment3)

# Display the segments using Matplotlib
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(puppy_segment1)
plt.title('Segment 1')
plt.axis('off')
plt.subplot(1, 3, 2)
plt.imshow(puppy_segment2)
plt.title('Segment 2')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(puppy_segment3)
plt.title('Segment 3')
plt.axis('off')

plt.show()
```



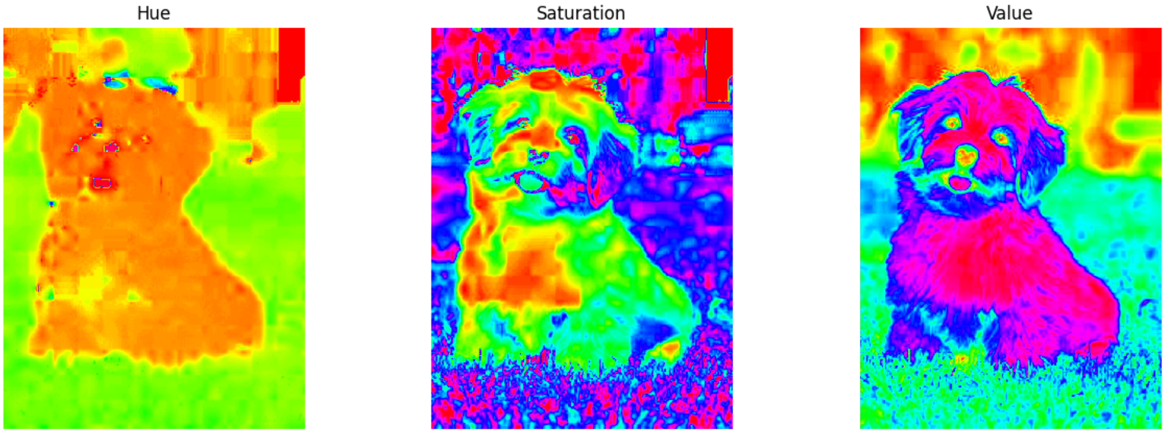
In [22]:

```
# Open the image
image = Image.open(r"C:\Users\navin\Downloads\puppy.jpg")

# Convert the image to numpy array
image_array = np.array(image)

# Split the image into its red, green, and blue channels
red_channel = image_array[:, :, 0]
green_channel = image_array[:, :, 1]
blue_channel = image_array[:, :, 2]

# Create a figure and axes for displaying the images horizontally
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Display the red, green, and blue channels
axes[0].imshow(red_channel, cmap='Reds')
axes[0].set_title('Red Channel')

axes[1].imshow(green_channel, cmap='Greens')
axes[1].set_title('Green Channel')

axes[2].imshow(blue_channel, cmap='Blues')
axes[2].set_title('Blue Channel')

# Hide the axes
for ax in axes:
    ax.axis('off')

# Display the images
plt.show()
```



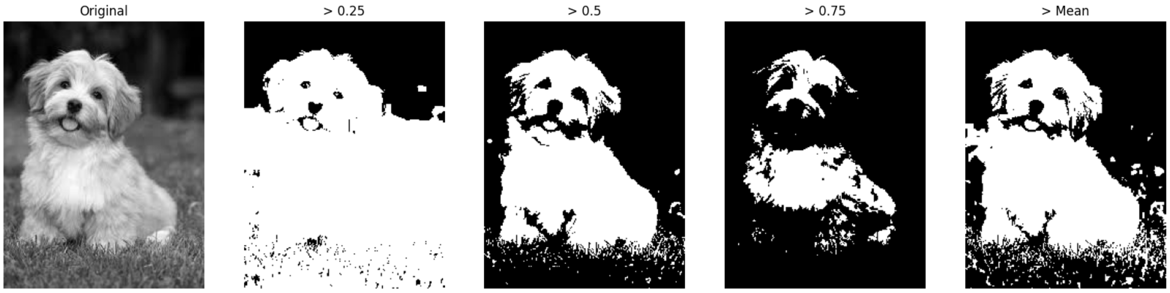
In [18]:

```
# Load the original image
original_image = cv2.imread(r"C:\Users\navin\Downloads\puppy.jpg")

# Convert the image to HSV color space
hsv_image = cv2.cvtColor(original_image, cv2.COLOR_BGR2HSV)

# Split the HSV image into three channels: Hue, Saturation, and Value
hue, saturation, value = cv2.split(hsv_image)

# Display the three channels horizontally
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Display the Hue channel
axes[0].imshow(hue, cmap='hsv')
axes[0].set_title('Hue')

# Display the Saturation channel
axes[1].imshow(saturation, cmap='hsv')
axes[1].set_title('Saturation')

# Display the Value channel
axes[2].imshow(value, cmap='hsv')
axes[2].set_title('Value')

# Remove the axis labels
for ax in axes:
    ax.axis('off')

# Show the plot
plt.show()
```



In [19]:

```
# Open the grayscale image
original_image = Image.open(r"C:\Users\navin\Downloads\puppy.jpg").convert("L")

# Convert the image to a numpy array
image_array = np.array(original_image)

# Define thresholds
thresholds = [0.25, 0.50, 0.75]

# Calculate mean of the image
image_mean = np.mean(image_array) / 255.0

# Create binary images based on thresholds
binary_images = [(image_array > threshold * 255).astype(np.uint8) for threshold

# Create greater than mean binary image
greater_than_mean = (image_array > image_mean * 255).astype(np.uint8)

# Combine all binary images into a list
all_images = [image_array] + binary_images + [greater_than_mean]

# Set up subplots
fig, axes = plt.subplots(1, len(all_images), figsize=(20, 5))

# Plot each image
for ax, img, title in zip(axes, all_images, ['Original'] + [f'> {t}' for t in th
    ax.imshow(img, cmap='gray')
    ax.set_title(title)
    ax.axis('off')

# Display the images
plt.show()
```



In [ ]: