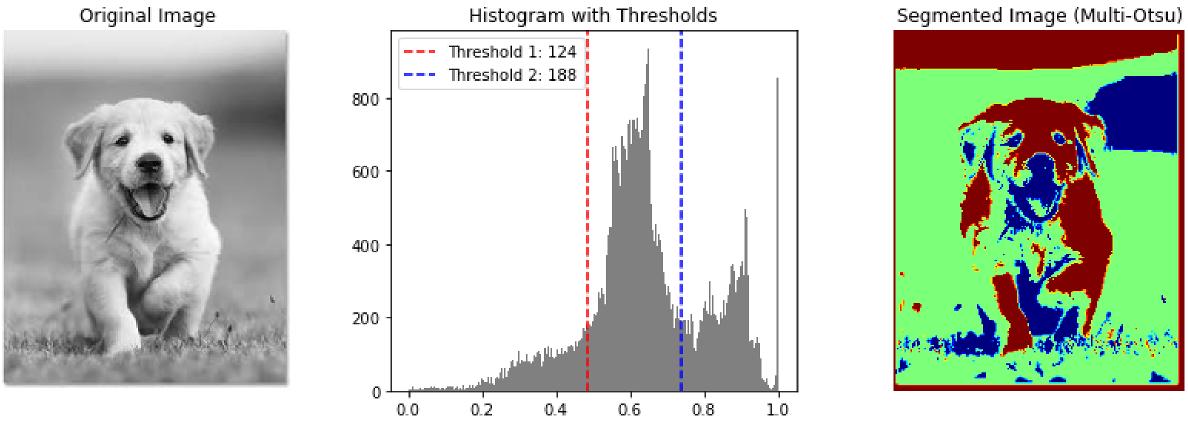
MULTIOTSU THRESHOLDING

```
import matplotlib.pyplot as plt
import numpy as np
from skimage import io
image = io.imread(r'C:\Users\dhanu\OneDrive\Desktop\dog.jpeg',as_gray=True)
hist, bin_edges = np.histogram(image.ravel(), bins=256, range=(0, 1))
bin_centers = (bin_edges[:-1] + bin_edges[1:]) / 2
total_pixels = image.size
sum = np.cumsum(hist)
mean = np.cumsum(hist * bin_centers)
global_mean = mean[-1] / total_pixels
max_variance = 0
best_thresholds = (0, 0)
for t1 in range(1, 256):
    for t2 in range(t1 + 1, 256):
        prob1 = sum[t1] / total_pixels
        prob2 = (sum[t2] - sum[t1]) / total_pixels
        prob3 = 1 - prob1 - prob2
        mean1 = mean[t1] / sum[t1] if sum[t1] != 0 else 0
        mean2 = (mean[t2] - mean[t1]) / (sum[t2] - sum[t1]) if (sum[t2] - sum[t1]) != 0 else 0
        mean3 = (mean[-1] - mean[t2]) / (total_pixels - sum[t2]) if (total_pixels - sum[t2]) != 0 else 0
        variance_between = (
            prob1 * (mean1 - global_mean) ** 2 +
            prob2 * (mean2 - global_mean) ** 2 +
            prob3 * (mean3 - global_mean) ** 2
        )
        if variance_between > max_variance:
            max_variance = variance_between
            best_thresholds = (t1, t2)
t1, t2 = best_thresholds
```

```
regions = np.digitize(image, bins=[t1 / 255, t2 / 255])

fig, ax = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))

ax[0].imshow(image, cmap='gray')

ax[0].set_title('Original Image')

ax[0].axis('off')

ax[1].hist(image.ravel(), bins=256, color='gray')

ax[1].axvline(t1 / 255, color='red', linestyle='--', label=f'Threshold 1: {t1}')

ax[1].axvline(t2 / 255, color='blue', linestyle='--', label=f'Threshold 2: {t2}')

ax[1].set_title('Histogram with Thresholds')

ax[1].legend()

ax[2].imshow(regions, cmap='jet')

ax[2].set_title('Segmented Image (Multi-Otsu)')

ax[2].axis('off')

plt.tight_layout()

plt.show()
```



Original Image

Histogram with Thresholds

Segmented Image (Multi-Otsu)