We first import the libraries which will be necessary for the lab task

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

Task-1

Explanation: We define a function that takes the image path, and a threshold value. The function loops over the pixel intensity values and compares it with the threshold value. If the value is higher the output binary image's pixel value will be 1, otherwise 0.

```
#Function which takes input path and threshold as input and returns a binary image
def to_binary(image_path, threshold):
  image=cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
  x, y = image.shape
  binary_image = np.zeros((x,y))
  for i in range(x):
    for j in range(y):
      if image[i][j]>threshold:
        binary_image[i][j] = 1
      else:
        binary_image[i][j] = 0
  return binary_image
```

```
#Loading the images as grayscale images
bright = cv2.imread('/content/bright.jpg', cv2.IMREAD_GRAYSCALE)
gray = cv2.imread('/content/gray.jpg', cv2.IMREAD_GRAYSCALE)
washed_out = cv2.imread('/content/washed_out.jpg', cv2.IMREAD_GRAYSCALE)
```

```
#Defining 3 threshold values
THRESHOLD_1 = 256/4 #First Quarter
THRESHOLD_2 = 256/2 #Middle Value
THRESHOLD_3 = (256/4)*3 #Third Quarter
```

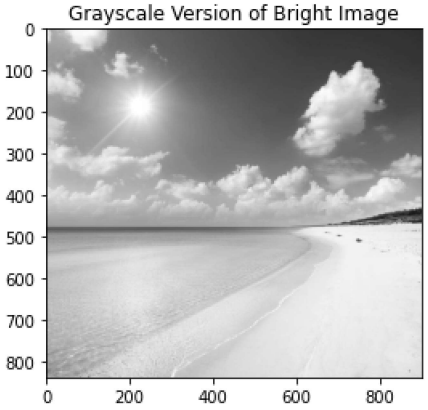
Now, we call the function on the loaded images using three threshold values

```
binary_bright1 = to_binary("/content/bright.jpg", THRESHOLD_1)
binary_gray1 = to_binary("/content/gray.jpg", THRESHOLD_1)
binary_washed1 = to_binary("/content/washed_out.jpg", THRESHOLD_1)
```

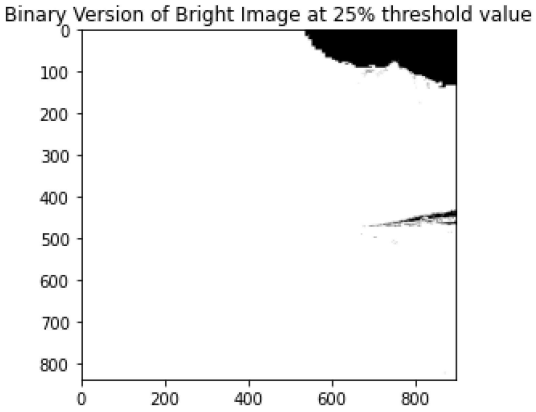
```
binary_bright2 = to_binary("/content/bright.jpg", THRESHOLD_2)
binary_gray2 = to_binary("/content/gray.jpg", THRESHOLD_2)
binary_washed2 = to_binary("/content/washed_out.jpg", THRESHOLD_2)
```

```
binary_bright3 = to_binary("/content/bright.jpg", THRESHOLD_3)
binary_gray3 = to_binary("/content/gray.jpg", THRESHOLD_3)
binary_washed3 = to_binary("/content/washed_out.jpg", THRESHOLD_3)
```

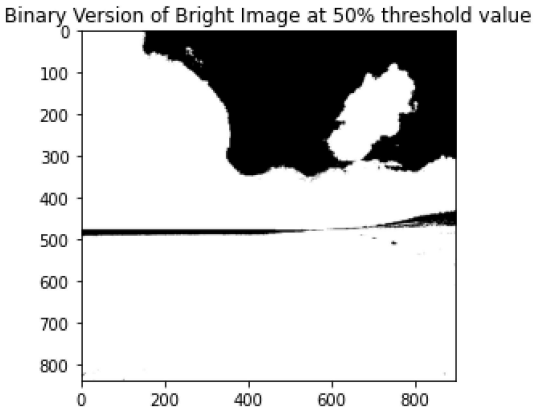
```
image=cv2.imread("/content/bright.jpg", cv2.IMREAD_GRAYSCALE)
plt.imshow(image, cmap = "gray")
plt.title("Grayscale Version of Bright Image")
plt.show()
```



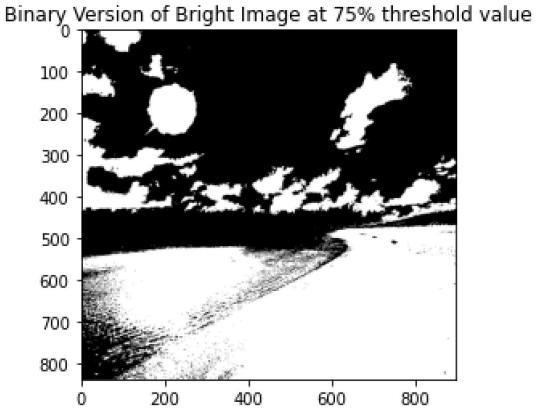

```
plt.imshow(binary_bright1, cmap = "gray")
plt.title("Binary Version of Bright Image at 25% threshold value")
plt.show()
```




```
plt.imshow(binary_bright2, cmap = "gray")
plt.title("Binary Version of Bright Image at 50% threshold value")
plt.show()
```

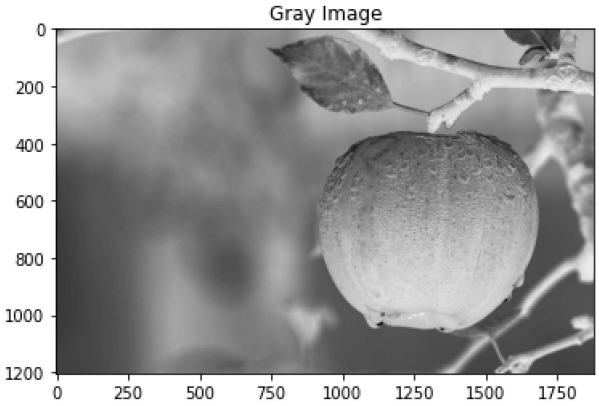



```
plt.imshow(binary_bright3, cmap = "gray")
plt.title("Binary Version of Bright Image at 75% threshold value")
plt.show()
```

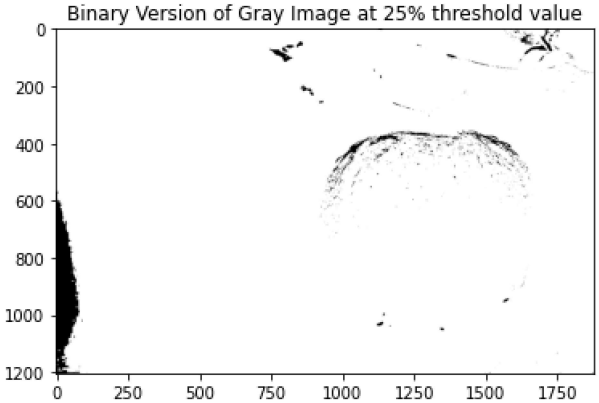


Explanation: At 25% threshold value, we lose most of the distinctive features of the image, such as, seperation of the sky with the sea. At 50%, the seperation still exists but no other detail is present. At 75%, we can see the best version where the details of the cloud, texture of the sea and sand is visible.

```
image_gray =cv2.imread("/content/gray.jpg", cv2.IMREAD_GRAYSCALE)
plt.imshow(image_gray, cmap = "gray")
plt.title("Gray Image")
plt.show()
```

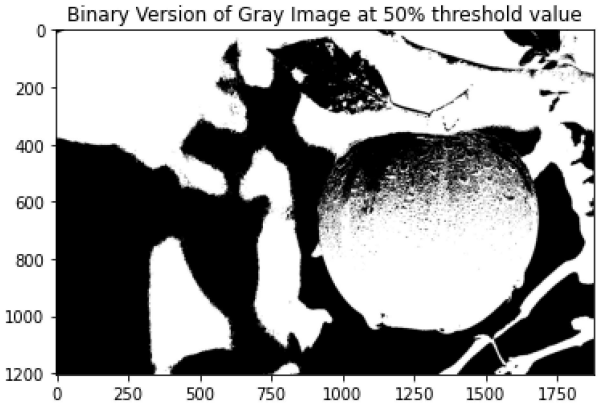



```
plt.imshow(binary_gray1, cmap = "gray")
plt.title("Binary Version of Gray Image at 25% threshold value")
plt.show()
```

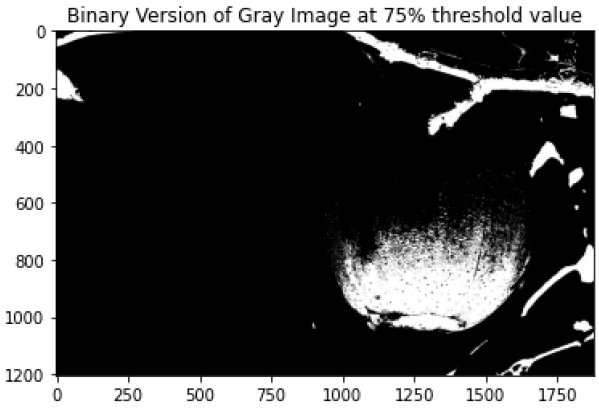



```
plt.imshow(binary_gray2, cmap = "gray")
```

```
plt.title("Binary Version of Gray Image at 50% threshold value")
plt.show()
```

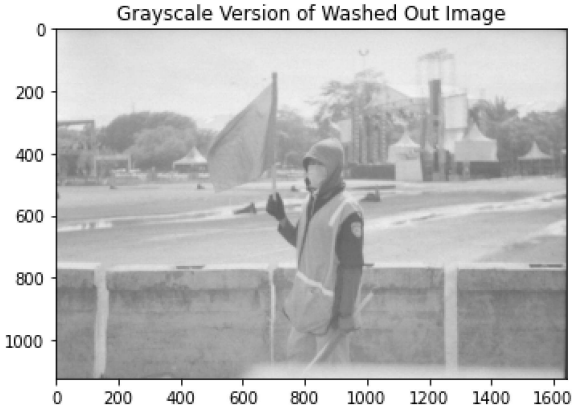



```
plt.imshow(binary_gray3, cmap = "gray")
plt.title("Binary Version of Gray Image at 75% threshold value")
plt.show()
```

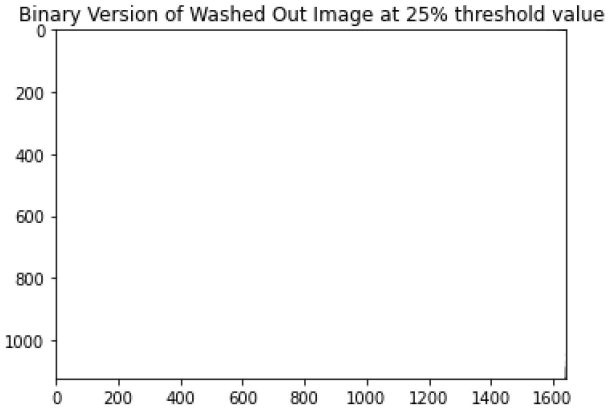


Explanation: Similar to the first image, we lose most of the distinctive features of the image at 25% of the threshold value. We find the best results at 50%. which shows clear distiction between the apple and the background. At 75%, more detail about the apple is visible but the background is indistinctable. As the threshold value is set to a higher value, most of the pixels are colored black. Since, the original image is relatively darker, at 75% value, it shows relatively poor results.

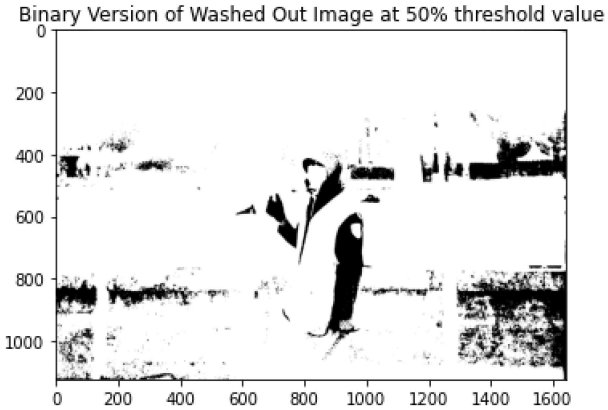
```
image_washed =cv2.imread("/content/washed_out.jpg", cv2.IMREAD_GRAYSCALE)
plt.imshow(image_washed, cmap = "gray")
plt.title("Grayscale Version of Washed Out Image")
plt.show()
```



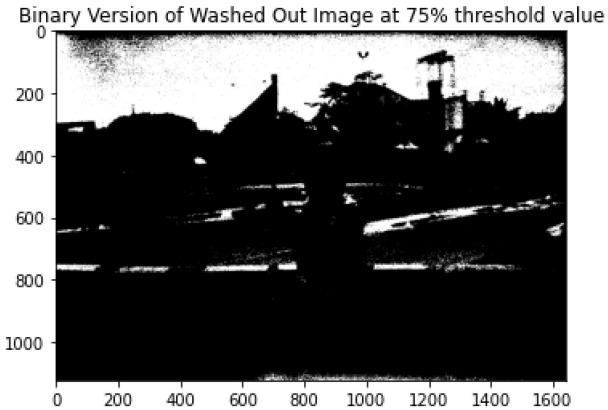
```
plt.imshow(binary_washed1, cmap = "gray")
plt.title("Binary Version of Washed Out Image at 25% threshold value")
plt.show()
```




```
plt.imshow(binary_washed2, cmap = "gray")
plt.title("Binary Version of Washed Out Image at 50% threshold value")
plt.show()
```




```
plt.imshow(binary_washed3, cmap = "gray")
plt.title("Binary Version of Washed Out Image at 75% threshold value")
plt.show()
```



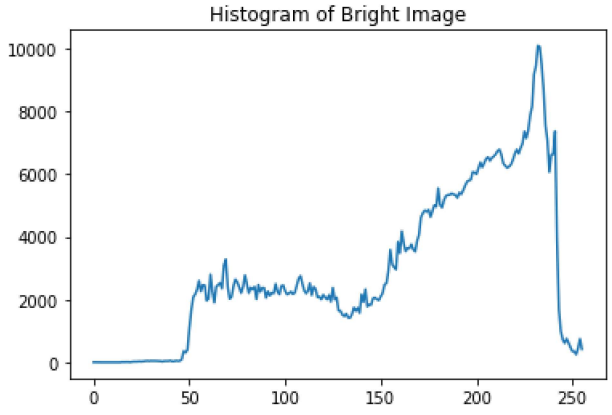
Explanation: As it is a washed out image, at 25% value, the image is completely white as all the pixels are colored white since the threshold value is low, and the lower values are set to 0. At 50% and 75%, we get similar details which show some

distinctive features.

Task-2

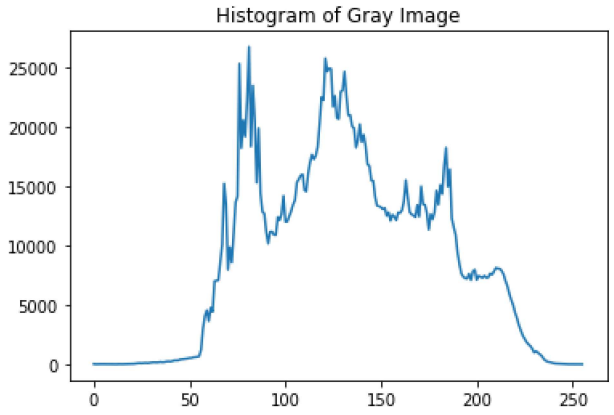
```
# calculate frequency of pixels in range 0-255
hist_bright = cv2.calcHist([bright],[0],None,[256],[0,256])
hist_gray = cv2.calcHist([gray],[0],None,[256],[0,256])
hist_washed = cv2.calcHist([washed_out],[0],None,[256],[0,256])
```

```
plt.plot(hist_bright)
plt.title('Histogram of Bright Image')
plt.show()
```



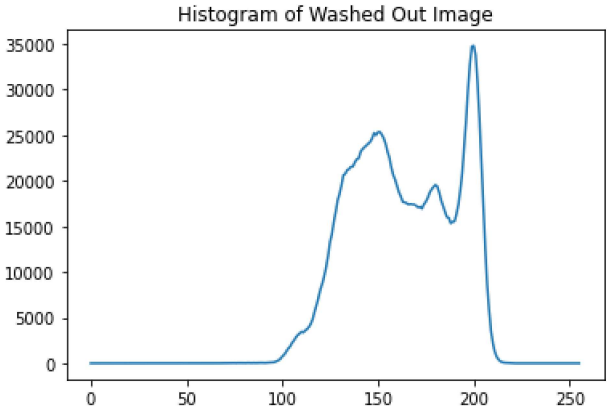
Explanation: From the histogram, we can see a higher distribution of brighter values and can conclude that the image is a relatively birghter image.

```
plt.plot(hist_gray)
plt.title('Histogram of Gray Image')
plt.show()
```



Explanation: From the histogram, we can tell the pixel values are almost evenly distributed. The image captured details in the whole spectrum.

```
plt.plot(hist_washed)
plt.title('Histogram of Washed Out Image')
plt.show()
```



Explanation: From the histogram, we can see the distribution is rather smoothened which means it is a washed out image. The distribution is more dense towards higher values implying that it a relatively bright image.