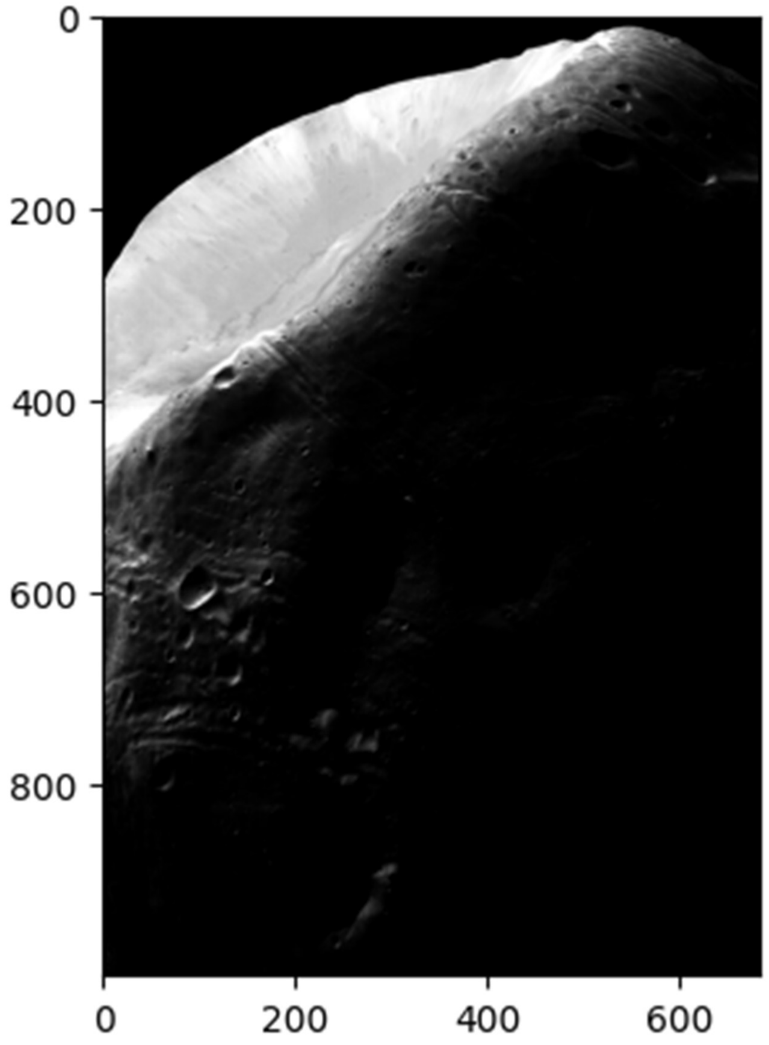
Code :

```
#Loads and display image
sample_image = cv2.imread(image_path+"Fig0323(a)(mars_moon_phobos).tif",0)
window_name = "sample"
show(sample_image,window_name)
plt.imshow(sample_image,cmap='gray')
```

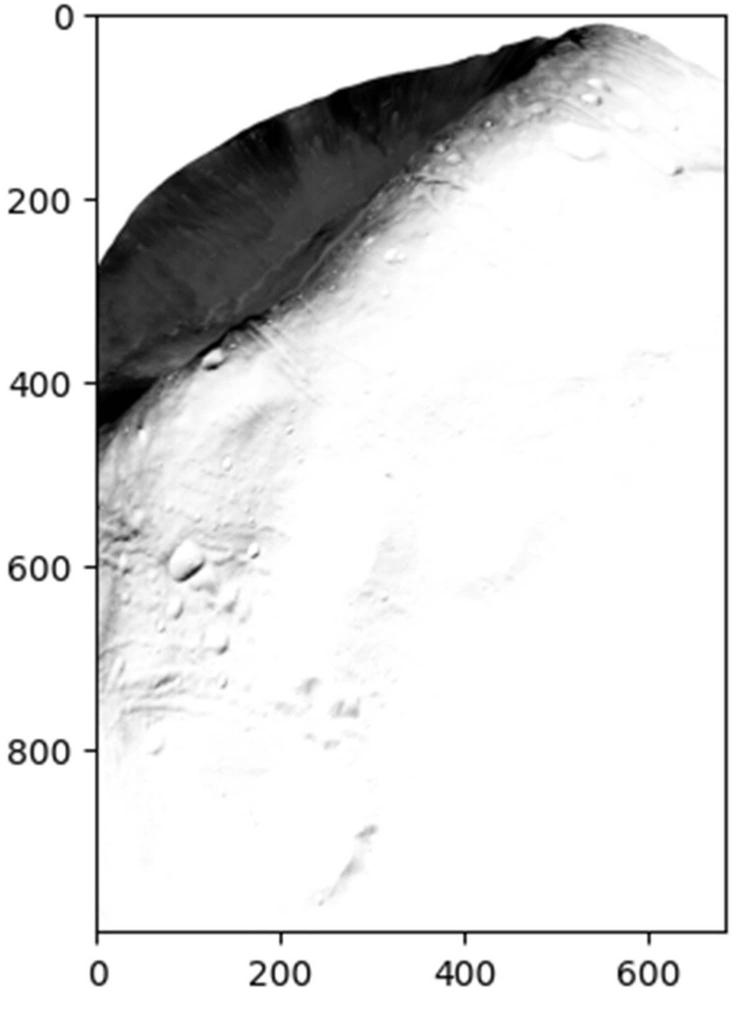
Output:



Code:

```
#negative image

def invert(image):
  nega_image = image.copy()
  nega_image = 255 - image # As image is a numpy array of type uint8, max value of pixel is
255
  return nega_image

nega_image = invert(sample_image)
show(nega_image)
plt.imshow(nega_image,cmap='gray')
```

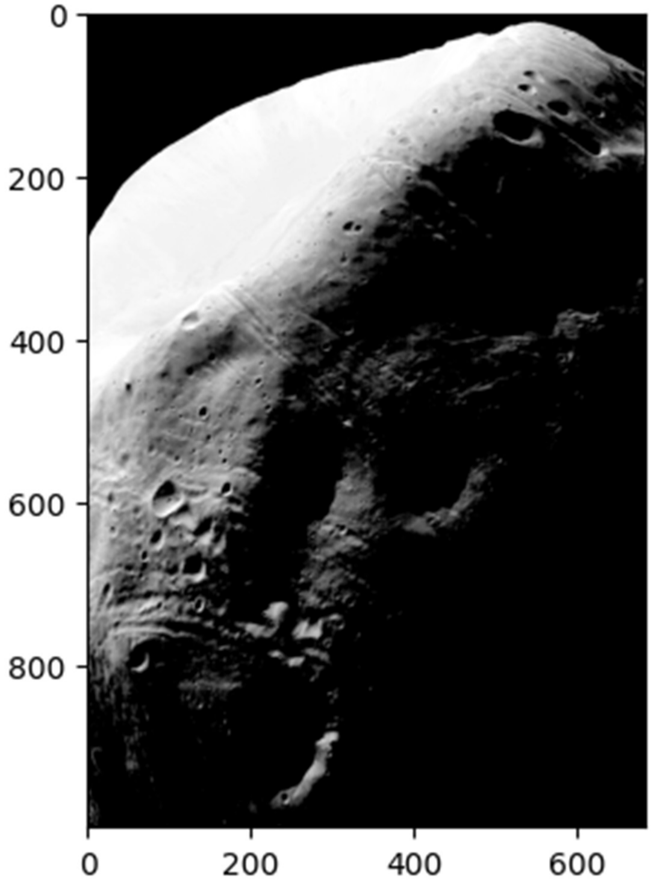
Output:



Code:

```
#log transform

def log_transform(image):
  new = image.copy()
  mask = image==0
  mask.astype(np.uint8) # create a mask to remove 0 values as log(0) -> -inf
  new = new + mask
  scale = 255/np.log(1+np.max(image))
  new = scale*np.log(new)
  new = np.asarray(new,np.uint8)
  return new


log_image = log_transform(sample_image)
show(log_image)
plt.imshow(log_image,cmap='gray')
```

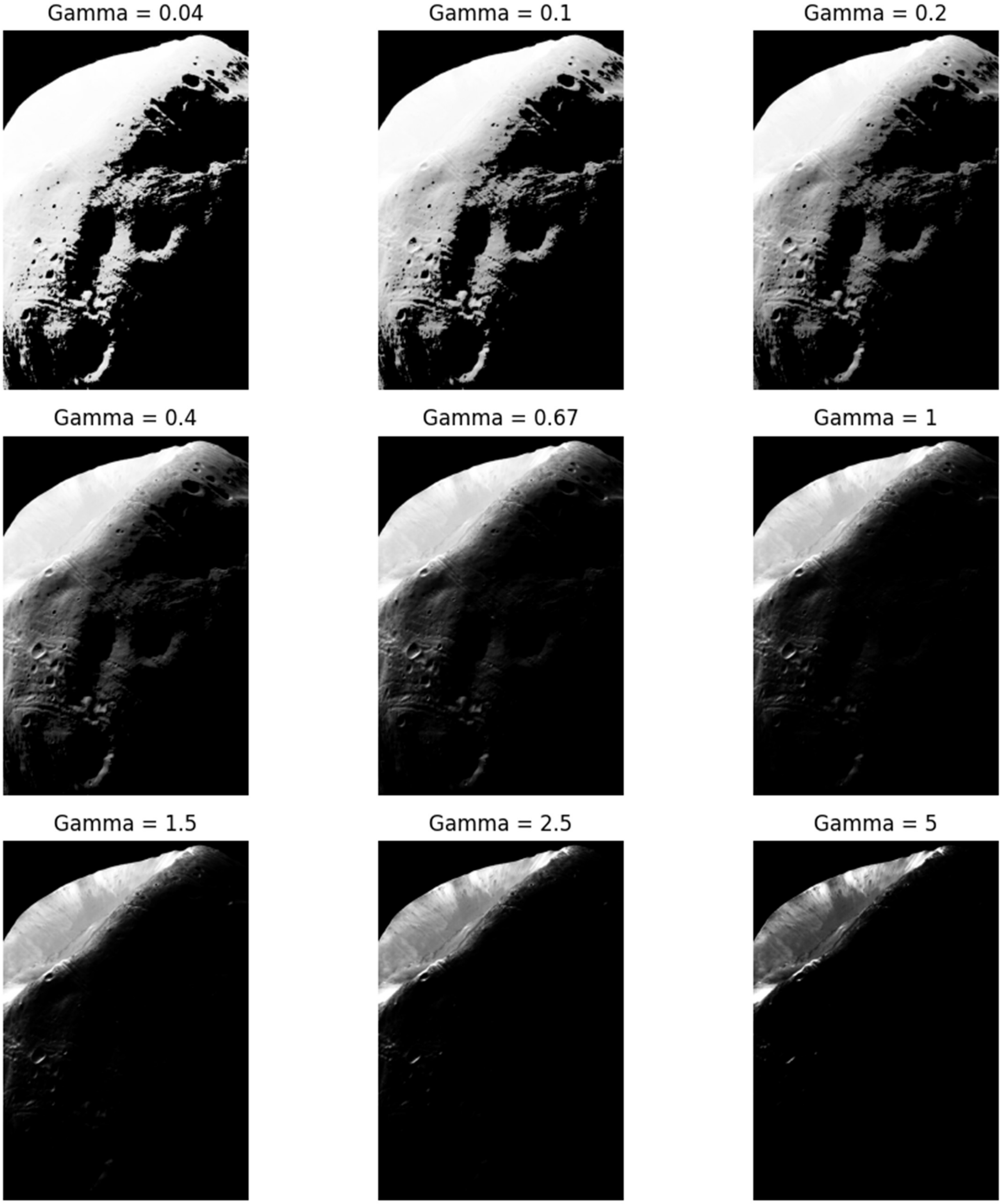
Output:



Code:

```
def gamma(r,image):
  new = image.copy()
  new = np.array(255*((image/255)**r),dtype=np.uint8)
  return new



gamma_images=[]
gamma_values = [0.04,0.1,0.2,0.4,0.67,1,1.5,2.5,5]
for gamma_vals in gamma_values:
  new = gamma(gamma_vals,sample_image)
  gamma_images.append(new)

rows = 3
cols = 3
fig, axes = plt.subplots(rows, cols, figsize=(10, 10))

for i in range(rows):
    for j in range(cols):
        index = i * cols + j
        axes[i, j].imshow(gamma_images[index], cmap='gray')
        axes[i, j].set_title(f'Gamma = {gamma_values[index]}')
        axes[i, j].axis('off')

plt.tight_layout()
plt.show()
```

Output:



Gamma = 0.04

Gamma = 0.1

Gamma = 0.2

Gamma = 0.4

Gamma = 0.67

Gamma = 1

Gamma = 1.5

Gamma = 2.5

Gamma = 5

Code:

```
#Image flip

def flip(image,orientation):
  new = image.copy()
  if orientation == 'h':
    new = cv2.flip(new,1)
  if orientation == 'v':
    new = cv2.flip(new,0)
  return new

rows = 1
cols = 2
fig, axes = plt.subplots(rows, cols, figsize=(10, 10))

axes[0].imshow(flip(sample_image,'v'), cmap='gray')
axes[0].set_title('Vertical')
axes[0].axis('off')

axes[1].imshow(flip(sample_image,'h'), cmap='gray')
axes[1].set_title('Horizontal')
axes[1].axis('off')


plt.tight_layout()
plt.show()
```

Output: