

```
# Function to subsample an image to a specific size
def subsample_image(image, target_size):
    return cv2.resize(image, target_size, interpolation=cv2.INTER_NEAREST)

# Function to resample an image to a specific size
def resample_image(image, target_size):
    return cv2.resize(image, target_size, interpolation=cv2.INTER_LINEAR)

# Function to display an image
def display_image(image, title):
    plt.imshow(image, cmap='gray')
    plt.title(title)
    plt.show()

original_image = cv2.imread('/content/Original_img.jpg', cv2.IMREAD_GRAYSCALE)

plt.imshow(original_image, cmap = "gray")
plt.title('Original Image (1024 x 1024)')
plt.axis("off")
plt.show()
```



```
# List of target sizes
target_sizes = [(512, 512), (256, 256), (128, 128), (64, 64), (32, 32)]

for target_size in target_sizes:
    # Subsample the original image to the current target size
    subsampled_img = subsample_image(original_image, target_size)

    # Resample the subsampled image back to the original size (1024x1024)
    resampled_img = resample_image(subsampled_img, (1024, 1024))

    # Plot the subsampled image
    plt.figure(figsize=(6, 6))
    plt.imshow(subsampled_img, cmap='gray')
    plt.title(f'Subsampled to {target_size[0]} x {target_size[1]}')
    plt.axis('off')
    plt.show()

    # Plot the resampled image
    plt.figure(figsize=(6, 6))
    plt.imshow(resampled_img, cmap='gray')
    plt.title(f'Resampled to {1024} x {1024}')
    plt.axis('off')
    plt.show()
```



Subsampled to 512 x 512



Resampled to 1024 x 1024



Subsampled to 256 x 256



Resampled to 1024 x 1024



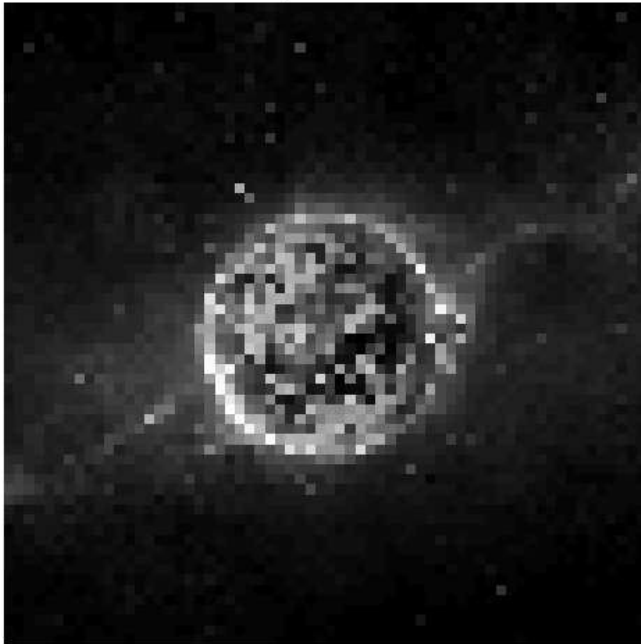
Subsampled to 128 x 128



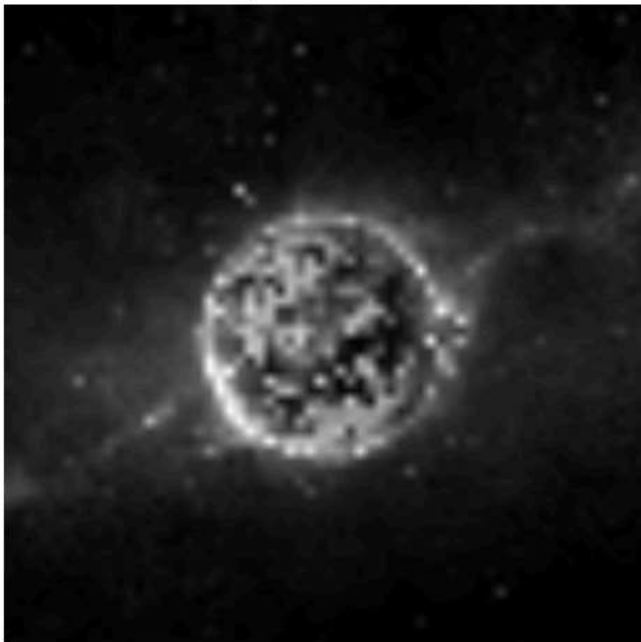
Resampled to 1024 x 1024



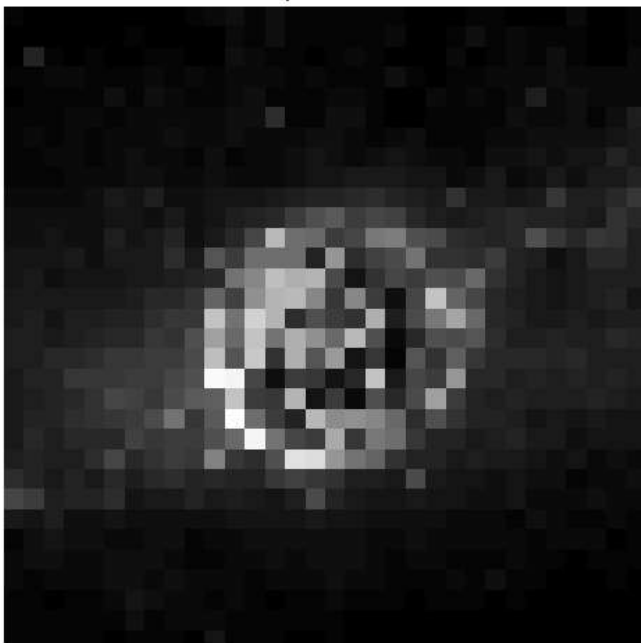
Subsampled to 64 x 64



Resampled to 1024 x 1024



Subsampled to 32 x 32



Resampled to 1024 x 1024

