

```

import cv2 as cv

def zoom_image(image, scale_factor):
    # Calculate new dimensions
    new_width = int(image.shape[1] * scale_factor)
    new_height = int(image.shape[0] * scale_factor)

    # Resize the image using nearest-neighbor interpolation
    zoomed_image = cv.resize(image, (new_width, new_height), interpolation=cv.INTER_NEAREST)

    return zoomed_image

# Read the input image
from google.colab import drive
drive.mount('/content/drive')

image = cv.imread("/content/drive/MyDrive/images/im01.png", cv.IMREAD_COLOR)

assert image is not None

# Check if the image is loaded successfully
if image is None:
    print("Error: Unable to load image.")
else:
    # Define the scaling factor (0 < s <= 10)
    scale_factor = 2.0

    # Check if the scale factor is within the valid range
    if scale_factor <= 0 or scale_factor > 10:
        print("Error: Invalid scale factor. Scale factor must be in the range (0, 10].")
    else:
        # Zoom the image
        zoomed_image = zoom_image(image, scale_factor)


# Display the original and zoomed images

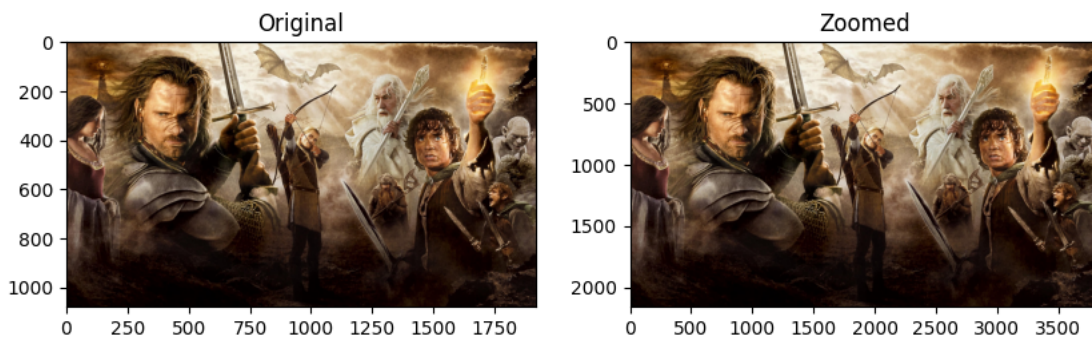
fig, ax = plt.subplots(1, 2, figsize = (10, 10))

ax[0].imshow(original_rgb)
ax[0].set_title('Original')
ax[1].imshow(zoomed_rgb)
ax[1].set_title('Zoomed')

plt.show()

```

 Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).



```
import cv2 as cv

def zoom_image(image, scale_factor):
    # Calculate new dimensions
    new_width = int(image.shape[1] * scale_factor)
    new_height = int(image.shape[0] * scale_factor)

    # Resize the image using bilinear interpolation
    zoomed_image = cv.resize(image, (new_width, new_height), interpolation=cv.INTER_LINEAR)

    return zoomed_image

# Read the input image
from google.colab import drive
drive.mount('/content/drive')

image = cv.imread("/content/drive/MyDrive/images/im01.png", cv.IMREAD_COLOR)

assert image is not None

# Check if the image is loaded successfully
if image is None:
    print("Error: Unable to load image.")
else:
    # Define the scaling factor (0 < s <= 10)
    scale_factor = 2.0

    # Check if the scale factor is within the valid range
    if scale_factor <= 0 or scale_factor > 10:
        print("Error: Invalid scale factor. Scale factor must be in the range (0, 10].")
    else:
        # Zoom the image using bilinear interpolation
        zoomed_image = zoom_image(image, scale_factor)

# Display the original and zoomed images

fig, ax = plt.subplots(1, 2, figsize = (10, 10))

ax[0].imshow(original_rgb)
ax[0].set_title('Original')
ax[1].imshow(zoomed_rgb)
ax[1].set_title('Zoomed')

plt.show()
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

