

Resizing and Blurring

Digital images are often very large; a higher resolution means a better image. We often find ourselves with a need to make the images smaller. One method to make the images smaller is to use every other pixel from the image. In a sense, we take a “sample” of the pixels from the original image. The following function, `quarter`, creates an image $\frac{1}{4}$ th the size of the original.

Example: Quarter function

```
# Creates a new picture 1/4 the size of
# the original
def quarter(picture):
    # create a new image
    width = picture.width//2
    height = picture.height//2
    newPic = Image.new('RGB', (width, height))

    # use every other row, every other column
    for y in range(0, picture.height, 2):
        for x in range(0, picture.width, 2):
            rvalue, gvalue, bvalue = picture.getpixel((x,y))
            newPic.putpixel((x//2, y//2), (rvalue, gvalue, bvalue))

    # return the new picture
    return newPic
```

Making an image larger (scaling up) is a little trickier, since we need to add information to the image. We will do this by taking every pixel twice. The following function, `quadruple`, creates an image 4 times the size of the original.

Example: Quadruple function

```
# Creates a new image 4 times the size
# of the original (2*width, 2*height)
def quadruple(picture):
    # create a new image
    width = picture.width*2
    height = picture.height*2
    newPic = Image.new('RGB', (width, height))

    for y in range(0, height):
        for x in range(0, width):
            # use each value of x twice for each y
```

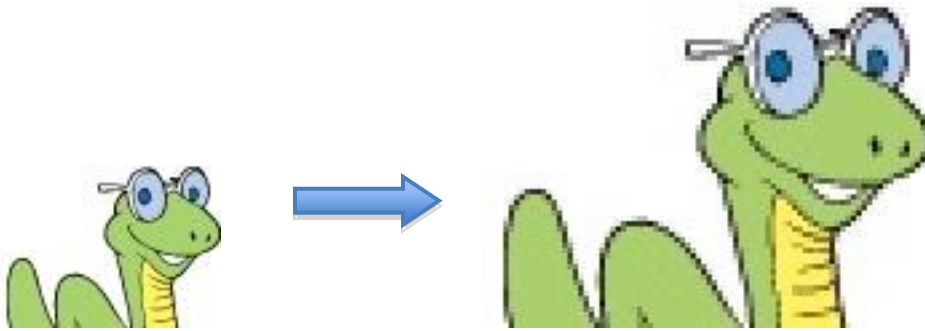
```

rvalue, gvalue, bvalue = picture.getpixel((x//2,y//2))
newPic.putpixel((x, y),(rvalue, gvalue, bvalue))

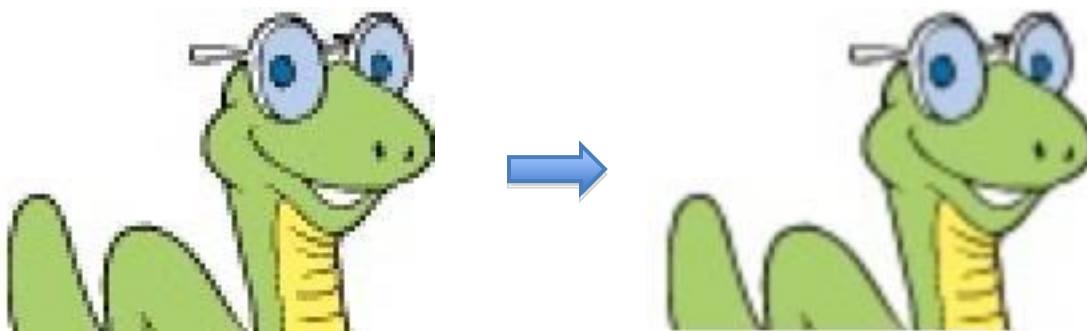
# return the new picture
return newPic

```

This function takes a picture as a parameter, and then makes a new picture that is twice as wide and twice as high as the original. We then loop through the x- and y-values of the new picture, and set the colors in the new picture based on the colors in the original picture. Notice, for instance, that pixels (0, 0), (1, 0), (0, 1), and (1, 1) in the new picture all get their color from pixel (0, 0) in the original picture. Pixels (2, 0), (3, 0), (2, 1), (3, 1) in the new picture get their color from pixel (1, 0) in the original picture. Each pixel in the original picture provides the color for 4 pixels in the new picture.



As we can see in the example, when we make pictures larger, we often end up with rough edges. These sharp steps in lines are called *pixelation*. To reduce this, we can blur the image – make the edges smoother on purpose.



Large picture

Blurred large picture

Notice in these pictures that some of the pixelation disappears in the blurry image. There are many algorithms for blurring pictures; we will use a fairly simple one. To blur a picture, we will set the color of each pixel to be the average of the red, green, and blue values of the pixels around it. Our function looks like this:

```

# A simple blur
def blur(picture):
    width = picture.width
    height = picture.height
    newPic = Image.new('RGB', (width, height))
    for y in range(1, height-1):
        for x in range(1, width-1):
            rtop, gtop, btop = picture.getpixel((x, y-1))
            rleft, gleft, bleft = picture.getpixel((x-1, y))
            rright, gright, bright = picture.getpixel((x+1, y))
            rbottom, gbottom, bbottom = picture.getpixel((x, y+1))
            rvalue, gvalue, bvalue = picture.getpixel((x, y))
            newR = (rtop + rleft + rright + rbottom + rvalue)//5
            newG = (gtop + gleft + gright + gbottom + gvalue)//5
            newB = (btop + bleft + bright + bbottom + bvalue)//5
            newPic.putpixel((x,y), (newR, newG, newB))

    return newPic

```

There are many other reasons to blur an image, including blurring backgrounds in a virtual meeting, blurring faces in an image to protect individuals' privacy, and blurring identifying information in an image such as addresses and license plate numbers.

You will explore resizing images and blurring images in the next activity.

Activity: Cropping and Resizing