## 0.1 Image cropping

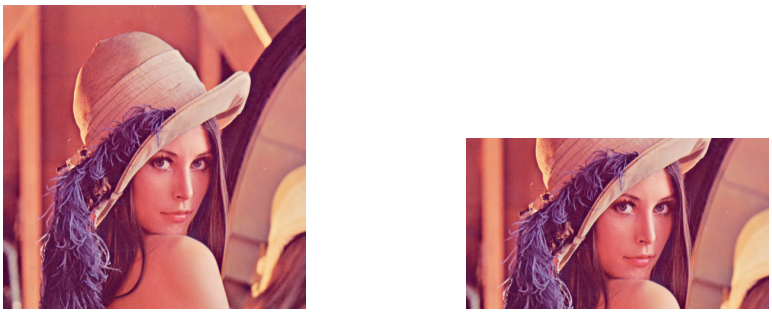
- *Algorithm:* Cropping (algo. 1)
- *Input:* Image to be cropped.
- *Complexity:* $\mathcal{O}(mn)$ depending on the size
- *Data structure compatibility:* Array.
- *Common applications:* Image processing.

**Problem.** Image cropping

Image cropping is an elementary operation on an image of removing an outer part of the image and leaving a smaller rectangular region. It can help remove the unwanted parts, improve image framing, highlight a part of the image or produce uniform datasets.

### Description

Image cropping is the removal of an outer part of the image. It can be performed physically as well as digitally. It would change the basic framing, ratio and structure of the image itself. The following Figure 1 gives an example of image cropping.



(a) Pic. 1 (b) Pic. 2

Figure 1: Group of pictures

The process of cropping a image is simple as explained literally. We will make this operation more mathematically accurate before discussing the algorithm. Suppose we have an bitmap image of size $a \times b$, and we hope to crop a $m \times n$-sized rectangular in the bottom-right part of the point $(x, y)$ from it. To reach this aim, just create a new array of size $m \times n$, and assign the corresponding color to each point. You may see the process in the pseudo-code in Algorithm 1.

**Algorithm 1:** Image Cropping, t

**Input :** Image $I$ as a 2-dim array of size $a \times b$, the target part size $m \times n$ and reference point $(x, y)$
**Output:** The new cropped image $T'$

```
I ← array of size m × n
for i ← 1to m do
    for j ← 1to n do
        I'[i][j] ← I[x + i][y + j]
    end for
end for
return I'
```

To assign value for each pixel in the $m \times n$ rectangular, it is easy to see that the algorithm has a time complexity of $\mathcal{O}(mn)$.