Digital Image Processing

Image Fundamentals

10th May, 2017

Image Scaling

- **Image scaling** is the process of resizing a digital image.

- Apart from fitting a smaller display area, image size is most commonly decreased (or **subsampled** or **downsampled**). Enlarging an image (**upsampling** or **interpolating**) is generally common for making smaller imagery fit a bigger screen in full screen mode, for example.

- In “zooming” an image, it is not possible to discover any more information in the image than already exists, and image quality inevitably suffers. However, there are several methods of increasing the number of pixels that an image contains, which evens out the appearance of the original pixels.

Image Scaling

- Zooming (up scaling, resizing upward) requires two steps
  - The Creation of new pixel locations
  - Assignment of gray levels to new pixel locations

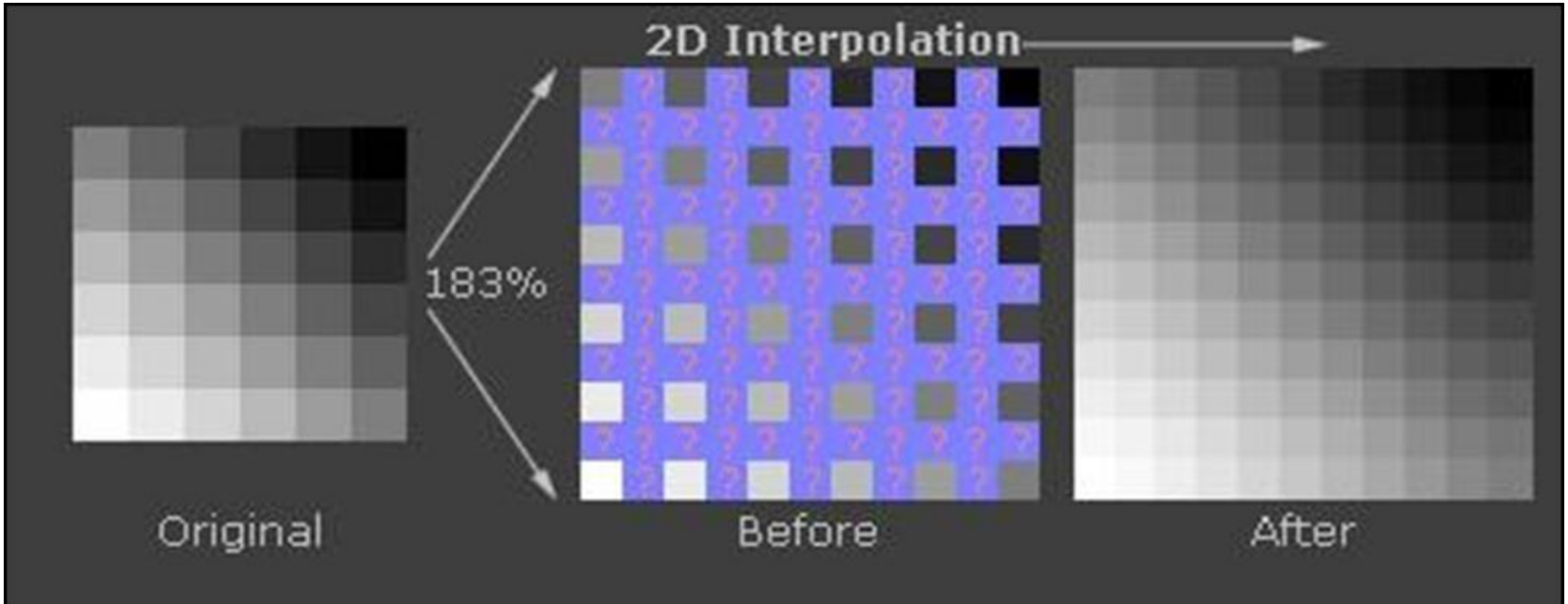


Image Scaling

- Zooming (up scaling, resizing upward) can be achieved by the following techniques:

  - **Nearest neighbor Interpolation**
  - **Pixel Replication (A variation of Nearest neighbor Interpolation)**
  - **Bilinear Interpolation**
  - **Bicubic Interpolation**

Nearest neighbor Interpolation

- Suppose that we have an image of size 500 x 500 and we want to enlarge it to 1.5 times 750 x 750 pixels.

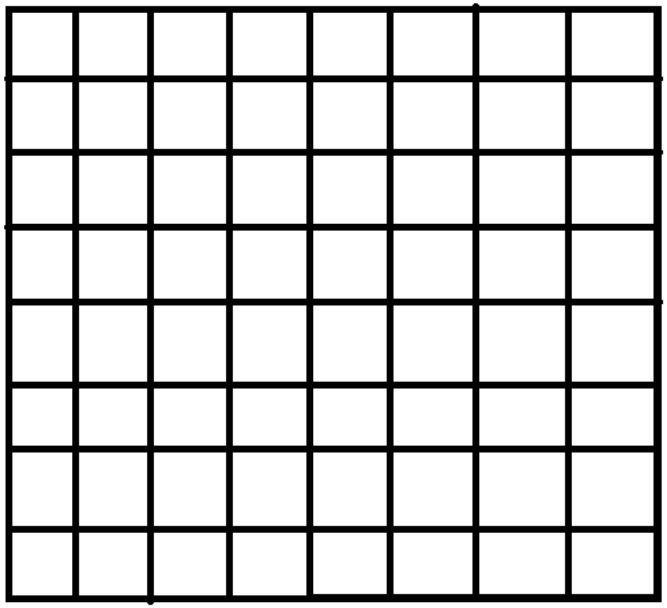
- For any zooming approach we have to create an imaginary grid of the size which is required over the original image. In that case we will have an imaginary grid of 750 x 750 over an original image.

- Obviously the spacing in the grid would be less than one pixel because we fitting it over a smaller image. In order to perform gray level assignment for any point in the overlay, we look for the closest pixel in the original image and assign its gray level to new pixel in the grid.

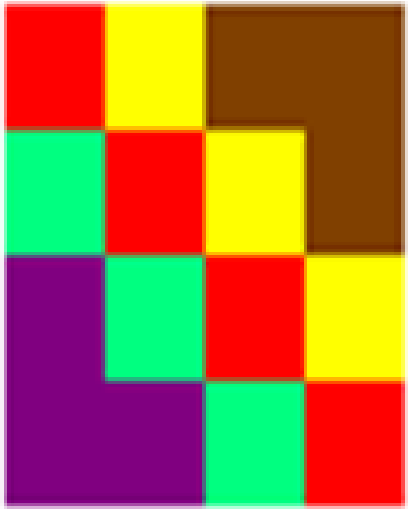
- When finished with all points in the grid, we can simply expand it to the originally specified size to obtain the zoomed image. This method of gray level assignment is called nearest neighbor interpolation

Nearest neighbor Interpolation

A 4x4 image which we want to scale up by a factor of 2



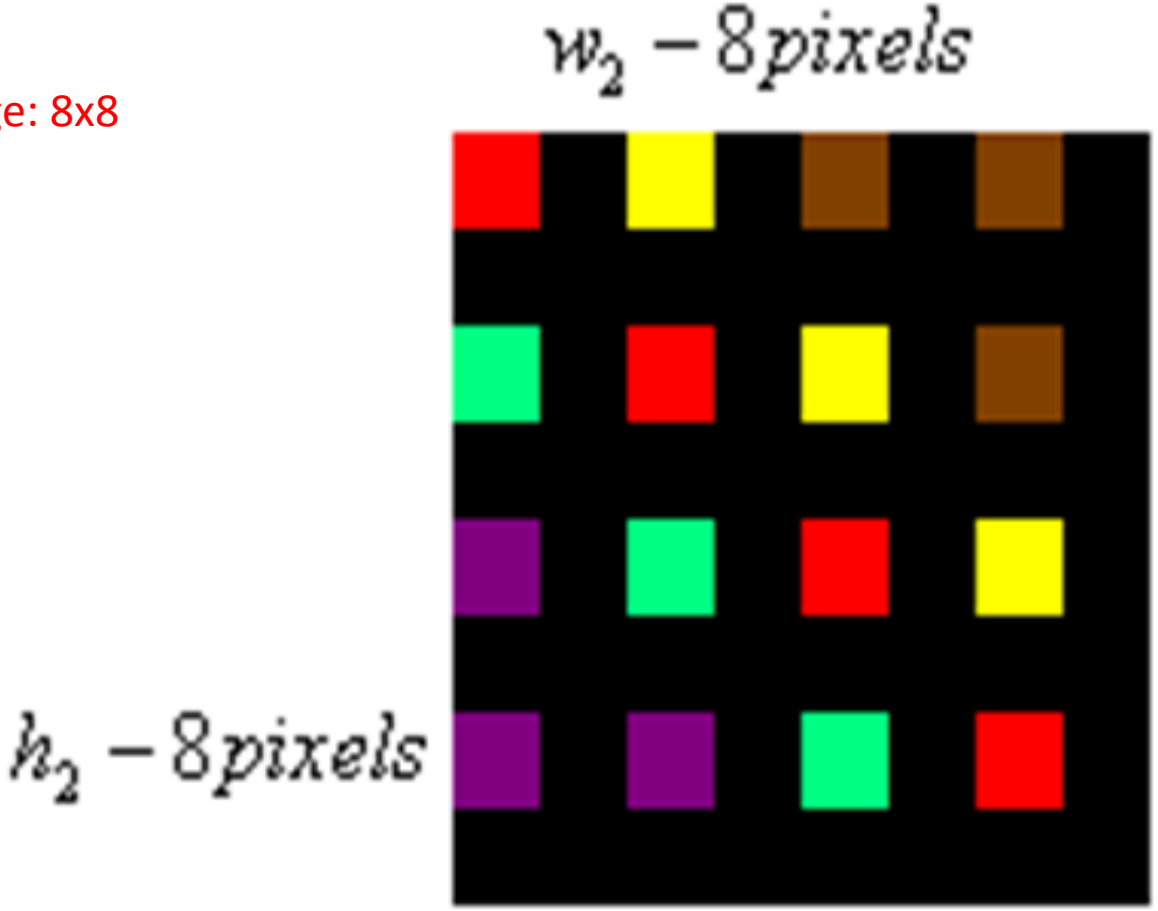
$w_1 - 4\,pixels$



$h_1 - 4\,pixels$

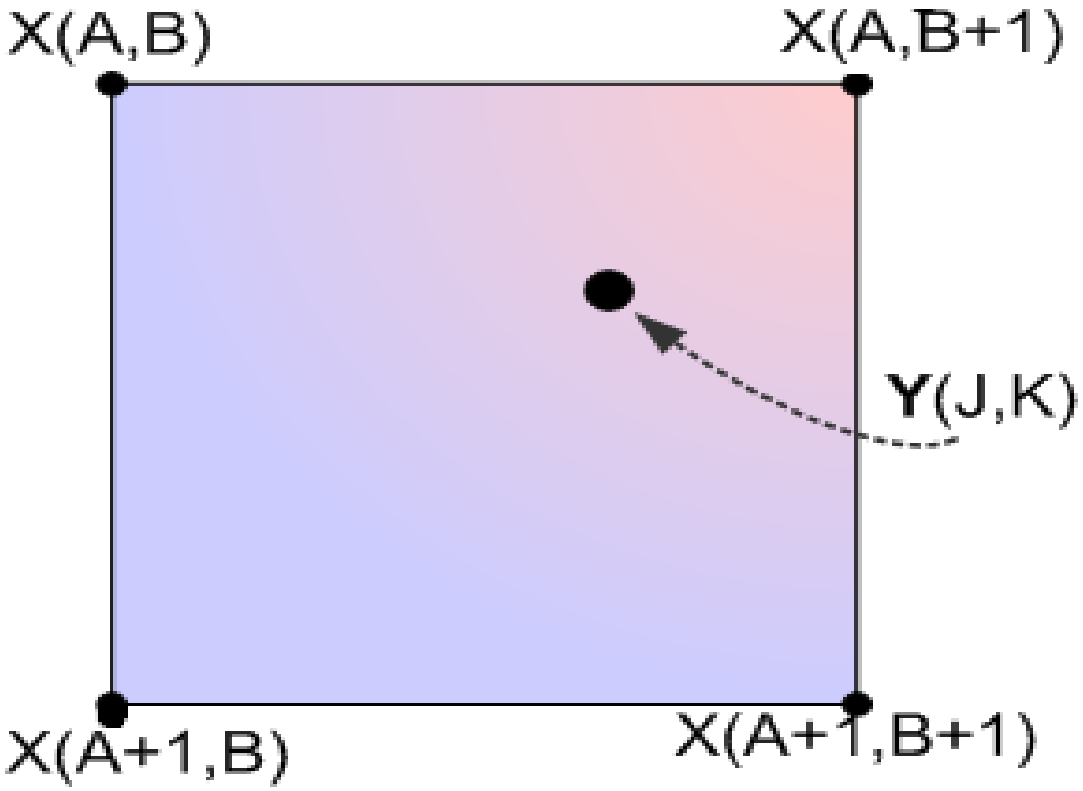
Nearest neighbor Interpolation

Scaled image: 8x8



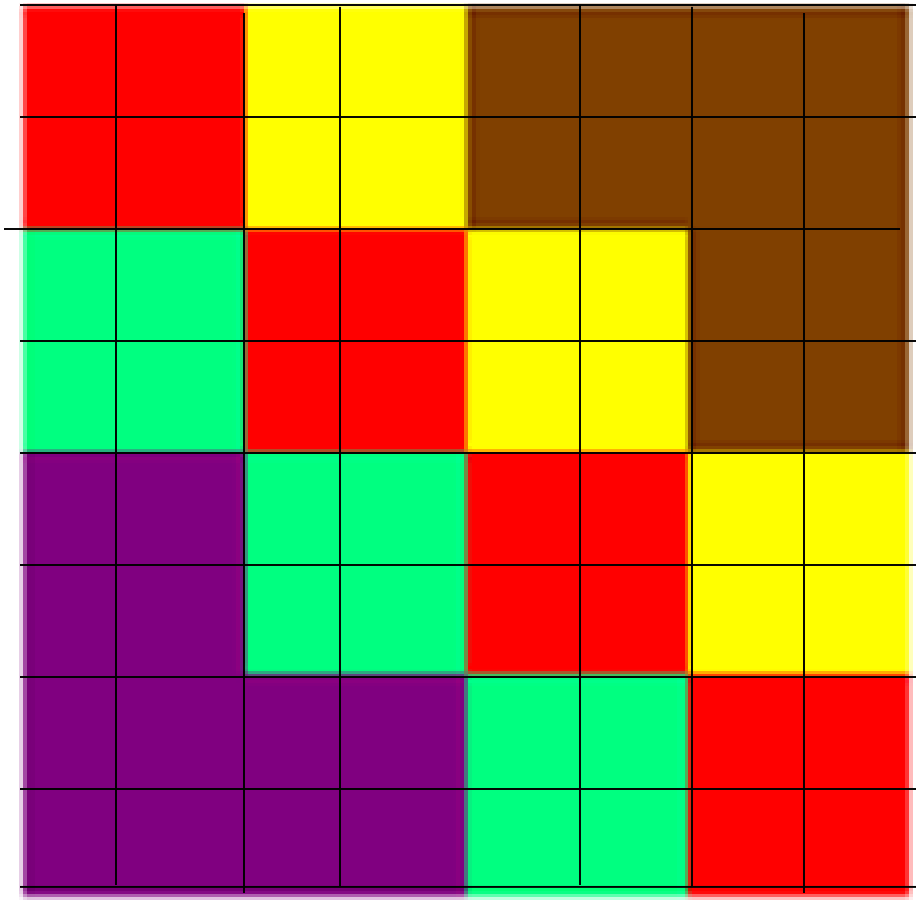
Nearest neighbor Interpolation

- Choosing the neighboring pixels



Nearest neighbor Interpolation

- Resultant Zoomed Image



Example

$$\begin{bmatrix} 69 & 50 & 80 \\ 45 & 60 & 66 \\ 30 & 55 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix}$$

Original image

image with rows expanded

image with rows and columns expanded

Matlab Command

- i=imread('cameraman.tif');
- y=imresize(i,[512,512],'nearest');
- imshow(i);
- figure,imshow(y);

Implementation in Matlab

```
a = imread('pout.tif');
[r,c] = size(a);
factor = 20;
[h, w, c] = size(a);
wn = w*factor;
hn = h*factor;
Img_zoomed = uint8(zeros(hn, wn));

for i= 0:hn-1
  for j= 0:wn-1
    x = floor(j/factor);
    y = floor(i/factor);
    for k= 1:c
      Img_zoomed(i+1, j+1, k) = a(y+1, x+1
    end
  end
end

figure, imshow(a);
figure,imshow(Img_zoomed);
```