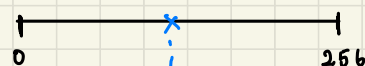


107070037

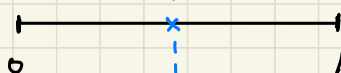
Project 02-02 reducing the number of intensity levels in an image

• Explanation

intensity : 256

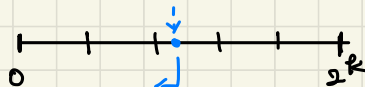


↓ normalize



↓ map 到 希望的 intensity

intensity : 2^k $k \in \{0, 1, 2, 3, 4, 5, 6, 7\}$



取整数 floor

在 imshow 的時候，標示 intensity 範圍

ex. `imshow(img, [0, intensity level - 1])`

Original Image



Reduce Intensity Level to 2

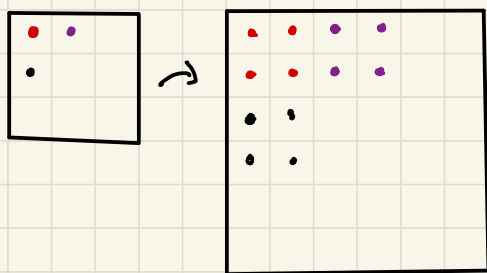


Project02-03 Zooming and Shrinking Images by Pixel Replication

• Explanation

Nearest - Neighbor algorithm

1. For zooming image :



放大 K 倍, 每个 row 和 col
就重复 K 次

① original : $m \times n$

放大 K 倍

② row - wise , col - wise pixel position
 $[1 : km]$ $[1 : kn]$

③ normalize , map to original img index

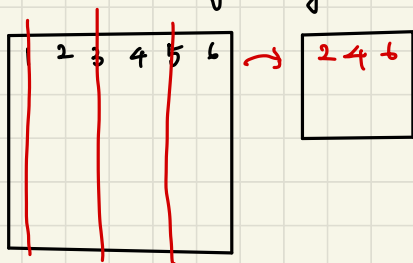
$$\text{ceil}\left(\frac{[1 : km]}{k}\right)$$

$$\text{ceil}\left(\frac{[1 : kn]}{k}\right)$$

④ perform interpolation by col-wise and row-wise

ex. First row = [10 20 30] column-wise position
= [1 1 2 2 3 3]

2. For shrinking image: (deletion)



做法和 1. 一樣

■ 當先 shrink image 時，會丟掉一些 pixel，
再放大回去，只是複製附近的值，
看起來比較殘缺尖銳

Original Image



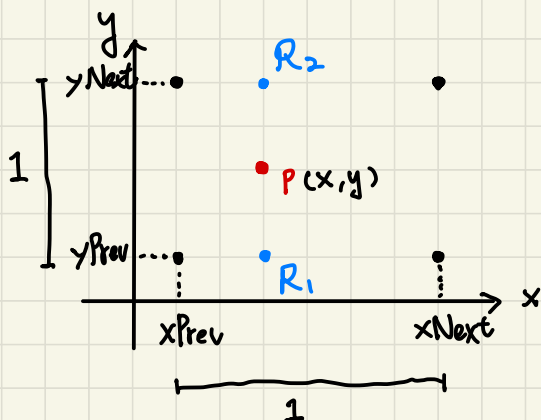
(NN)Shrink by factor
0.1



(NN)Zoom by factor
10



• Explanation



① 先做 x 方向 interpolation

$$f(R_1) = \frac{x_{Next} - x}{1} f(x_{Prev}, y_{Prev}) +$$

$$\frac{x - x_{Prev}}{1} f(x_{Next}, y_{Prev})$$

$$f(R_2) = \frac{x_{Next} - x}{1} f(x_{Prev}, y_{Next}) +$$

$$\frac{x - x_{Prev}}{1} f(x_{Next}, y_{Next})$$

② 做 y 方向 interpolation

$$f(p) = \frac{y_{Next} - y}{1} f(R_1) + \frac{y - y_{Prev}}{1} f(R_2)$$

做 bilinear interpolation, 先將放大縮小的圖片
或
上的 pixels map 回 original img 上, 注意使用
floor, max, min, 讓它不要超過圖片大小

■ 圖片 shrinking 後再 zoom 會使圖片模糊。

但比起用 Nearest Neighbor, 圖片會看起來
更自然, 比較不尖銳

Original Image



(Bilinear)Shrinking Img by
0.08



(Bilinear)Zooming Img by
12.5

