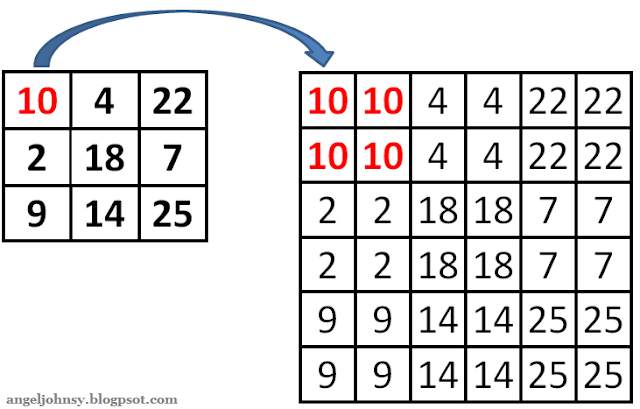
در این گزارش، خلاصه ای از الگوریتم بکار رفته جهت گسترش یا فشرده سازی تصویر بیان می کنیم.

در این کد که به زبان پایتون نوشته شده است از کتابخانه scikit-image (skimage) برای دریافت سیگنال های تصویر استفاده می کنیم.

ابتدا تصویر را به دو حالت رنگی و سیاه و سفید خوانده و با استفاده از روش Nearest Neighbor سایز آن را تغییر می دهیم.

روش Nearest Neighbor یکی از روش های Image Scaling بوده که در آن رنگ خانه های اضافه شده در مراحل تغییر سایز به رنگ خانه ای که نزدیکتر است تبدیل می شود.



**توضیحات کد**

ابتدا تصویر را به دو صورت رنگی و سیاه و سفید می خوانیم:

colored\_image = img\_as\_ubyte(io.imread("KimiaMahdinejad.jpg", as\_gray=False))  
gray\_image = img\_as\_ubyte(io.imread("KimiaMahdinejad.jpg", as\_gray=True))

سپس طول و عرض تصویر را ذخیره می کنیم:

width, height = colored\_image.shape[:2]

تابع resize را تشکیل می دهیم:

def resize\_nearest\_neighbor(image, ratio):  
 # Setting new width and height  
 new\_width = int(width / ratio)  
 new\_height = int(height / ratio)  
  
 # Calculating the scaling factor  
 width\_scale = new\_width / (width - 1)  
 height\_scale = new\_height / (height - 1)  
  
 # Creating a matrix new width and height  
 new\_image = np.zeros([new\_width, new\_height, 3])  
  
 # Creating new image based on nearest neighbor algorithm  
 for width\_index in range(new\_width - 1):  
 for height\_index in range(new\_height - 1):  
 new\_image[width\_index + 1, height\_index + 1] = image[1 + int(width\_index / width\_scale), 1 + int(height\_index / height\_scale)]  
  
 return new\_image

و در نهایت تابع را صدا می زنیم و نتیجه را دخیره می کنیم:

# X(2m, 2n)  
nearest\_neighbor\_resized\_colored\_2 = resize\_nearest\_neighbor(colored\_image, 2)  
io.imsave("KimiaMahdinejad\_NearestNeighbor\_Colored\_2.jpg", nearest\_neighbor\_resized\_colored\_2.astype(np.uint8))

nearest\_neighbor\_resized\_gray\_2 = resize\_nearest\_neighbor(gray\_image, 2)  
io.imsave("KimiaMahdinejad\_NearestNeighbor\_Gray\_2.jpg", nearest\_neighbor\_resized\_gray\_2.astype(np.uint8))  
  
# X(m/2, n/2)  
nearest\_neighbor\_resized\_colored\_1\_2 = resize\_nearest\_neighbor(colored\_image, 1/2)  
io.imsave("KimiaMahdinejad\_NearestNeighbor\_Colored\_1\_2.jpg", nearest\_neighbor\_resized\_colored\_1\_2.astype(np.uint8))  
  
nearest\_neighbor\_resized\_gray\_1\_2 = resize\_nearest\_neighbor(gray\_image, 1/2)  
io.imsave("KimiaMahdinejad\_NearestNeighbor\_Gray\_1\_2.jpg", nearest\_neighbor\_resized\_gray\_1\_2.astype(np.uint8))

**نتیجه**



تصویر اصلی



تصویر رنگی گسترده شده (X(m/2, n/2))



تصویر سیاه و سفید گسترده شده (X(m/2, n/2))



تصویر رنگی فشرده شده (X(2m, 2n))



تصویر سیاه و سفید فشرده شده (X(2m, 2n))