**Edge detection**

|  |
| --- |
| # -\*- coding: utf-8 -\*-  """  @author: ADMIN  """  **import** numpy **as** np  **import** cv2  # read image and convert to grayscale  img1 **=** cv2**.**imread**(**'anh.png'**,** 0**)**  # compute sobel-x  sobelx **=** cv2**.**Sobel**(**img1**,** cv2**.**CV\_64F**,** 1**,**0**)**  # compute sobely  sobely **=** cv2**.**Sobel**(**img1**,** cv2**.**CV\_64F**,** 0**,**1**)**  # Save results  cv2**.**imwrite**(**'vn\_edge\_x.png'**,** sobelx**)**  cv2**.**imwrite**(**'vn\_edge\_y.png'**,** sobely**)**  cv2**.**imwrite**(**'vn\_grayscale.png'**,** img1**)** |

**Interpolation: image upsampling.**

|  |
| --- |
| # -\*- coding: utf-8 -\*-  """  @author: ADMIN  """  **import** numpy **as** np  **import** cv2  # read a color image  img**=** cv2**.**imread**(**'anh.png'**)**  # get meta-data of the image  height**,** width**,** channels **=** img**.**shape  # new dimension  new\_dim **=** **(**width**\***4**,** height**\***4**)**  # upsampling the image  resize\_nearest **=** cv2**.**resize**(**img**,** new\_dim**,** interpolation **=** cv2**.**INTER\_NEAREST**)**  resize\_bilinear **=** cv2**.**resize**(**img**,** new\_dim**,** interpolation **=** cv2**.**INTER\_LINEAR**)**  # save result  cv2**.**imwrite**(**'resize\_nearest.png'**,** resize\_nearest**)**  cv2**.**imwrite**(**'resize\_bilinear.png'**,** resize\_bilinear**)** |