**1- Source codes in *"LinearAlgebra in image processing.pdf"***

**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**)** |
| # *Result after running*  *# Try to test also in JupyterLab* |

**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**)** |
| # *Result after running:*  *# Try to test also in JupyterLab* |

**Add Vector**

|  |
| --- |
|  |
| # *Result after running:*  *# Try to test also in JupyterLab* |

**Matrix Multiplication**

|  |
| --- |
|  |
| # *Result after running:*  *# Try to test also in JupyterLab* |

**Background Subtraction**

|  |
| --- |
| # *Skip due to lack of information:* |
| # *Result after running:*  *# Try to test also in JupyterLab* |

**Dot Product**

|  |
| --- |
|  |
| # *Result after running:*  *# Try to test also in JupyterLab* |

**Hadamard Product**

|  |
| --- |
|  |
| # *Result after running:*  *# Try to test also in JupyterLab* |

**Cosine Similarity**

|  |
| --- |
|  |
| # *Result after running:*  *# Try to test also in JupyterLab* |

**2- Library installation and study**

|  |  |  |  |
| --- | --- | --- | --- |
| **Library name** | **Command to install**  **&**  **Command for version check** | **Library roles** | **Links to references** |
| Numpy | *Pip install numpy*  *Pip list*  *python -m pip show numpy* | NumPy is the fundamental package for scientific computing in Python.  It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. | <https://cs231n.github.io/python-numpy-tutorial/>  <https://www.w3schools.com/python/numpy/default.asp>  <https://cs231n.github.io/python-numpy-tutorial/> |
| Pandas |  |  |  |
| Matplotlib |  |  |  |
| Keras |  |  |  |
| Torch |  |  |  |
| torchvision |  |  |  |
| opencv-python #for cv2 |  |  |  |
| pydub |  |  |  |
| tensorflow |  |  |  |
|  |  |  |  |