

IMP 301

Progress Test 1

Lê Đức Anh - Anhldhe150197 - AI1602

**Q1 (1.5pts) Implement some basic intensity transformation functions.**

**1.1 Negative transform**

| Source code | L**=**256  **def** T\_negative**(**f**):**  #do something here  g**=** L**-**1 **-**f  **return** g |
| --- | --- |
| Input | Output |
|  |  |

**1.2 Log transform (s=c log(1+r))**

| Source code | **def** logTransform**(**r**):**  c **=** 255**/(**np**.**log**(**1 **+** np**.max(**img**)))**  s **=** c**\***np**.**log**(**1.0**+**r**)**  **return** s |
| --- | --- |
| Input | Output |
|  | |

**1.3 Inverse Log transform**

| Source code | **def** inverseLogTransform**(**img**):**  c **=** 255**/(**np**.**log**(**1 **+** np**.max(**img**)))**  inv\_log\_transformed **=** np**.**exp**(**np**.**divide**(**img**,** c**))-**1  inv\_log\_transformed **=** np**.**array**(**inv\_log\_transformed**,** dtype **=** np**.**uint8**)**  **return** inv\_log\_transformed |
| --- | --- |
| Input | Output |
|  | |

**1.4 Power Law (Gamma) transformation**

| Source code | **def** gammaTransform**(**r**,** gamma**):**  c **=** 255**/(**np**.**log**(**1 **+** np**.max(**r**)))**  s **=** c**\*** **(**r**\*\***gamma**)**  **return** s |
| --- | --- |
| Input | Output |
| Trying with gamma < 1 [[0.6, 0.4, 0.3] | |
| Trying with gamma > 1 [3.0, 4.0, 5.0] | |

**1.5 and 1.6 ( nth root and nth root power)**

| Source code | **def** nRoot**(**input\_img**,** n**):**  **return** input\_img **\*\*** **(**1 **/** n**)** |
| --- | --- |
| Input | Output |
| with n = 2 | |
| nth root power with n = 0.5 | |

**Q2 (1.5pts) Implement histogram equalization**

| Source code | **def** T\_Histogram**(**img**,** L **=** 256 **):**  #Initialize intensity values with 256 zeroes  intensity\_count **=** **[**0**]** **\*** 256  height**,**width **=** img**.**shape**[:**2**]**  N **=** height **\*** width  high\_contrast **=** np**.**zeros**(**img**.**shape**)** #Array for new\_image  **for** i **in** **range(**0**,**height**):**  **for** j **in** **range(**0**,**width**):**  intensity\_count**[**img**[**i**][**j**]]** **+=** 1 #Find pixels count for each intensity  intensity\_count**,**total\_values\_used **=** np**.**histogram**(**img**.**flatten**(),**L**,[**0**,**L**])**  pdf\_list **=** np**.**ceil**(**intensity\_count**\*(**L**-**1**)/**img**.**size**)** #Calculate PDF  cdf\_list **=** pdf\_list**.**cumsum**()** #Calculate CDF  **for** y **in** **range(**0**,** height**):**  **for** x **in** **range(**0**,** width**):**  #Apply the new intensities in our new image  high\_contrast**[**y**,**x**]** **=** cdf\_list**[**img**[**y**,**x**]]**  **return** high\_contrast |
| --- | --- |
| Input | Output |
|  | |

**Q3 (2.5pts) Implement mean filter with zero-padding and**

| Source code | **def** meanFilter**(**image**,** string**):**  kernel\_size **=** 3  kernel **=** np**.**ones**((**kernel\_size**,** kernel\_size**),** dtype**=**"float"**)** **\*** **(**0.5 **/** **(**kernel\_size **\*** kernel\_size**))**  kernel **=** np**.**flipud**(**np**.**fliplr**(**kernel**))**  padImage**=**np**.**pad**(**image**,((**1**,** 1**),** **(**1**,** 1**)),** string**)**  **filter** **=** scipy**.**signal**.**convolve2d**(**padImage**,** kernel**,** mode**=**'same'**,** boundary**=**'fill'**,** fillvalue**=**0**)**  **return** **filter** |
| --- | --- |
| Input | Output |
| Nearest neighbor | |
| Zero padding | |

**Q4 (2.5pts) Implement 7x7 median filter.**

**Using: XrayCircuitBoardWithSaltnPepperNoise.png**

| Source code | **import** cv2  **from** google**.**colab**.**patches **import** cv2\_imshow  **from** PIL **import** Image  fname**=**"/content/drive/MyDrive/IMP301/PT1/images\_PT1/XrayCircuitBoardWithSaltnPepperNoise.png"  image **=** cv2**.**imread**(**fname**)**  processed\_image **=** cv2**.**medianBlur**(**image**,** 7**)**  temp **=** cv2**.**hconcat**([**image**,** processed\_image**])**  cv2\_imshow**(**temp**)** |
| --- | --- |
| Input | Output |
|  | |