IPCV Practical No :1

09:10

Aim

Implementation of Point Processing Techniques (Digital Negative, Thresholding, Intensity Transformation, Contrast Stretching).

Requirements

Operating System: Windows/Linux

Software: Python 3.x, Jupyter Notebook / Google Colab

Libraries: OpenCV, NumPy, Matplotlib

Theory

Point processing techniques are the **simplest image enhancement methods** where the new pixel value depends **only on the corresponding input pixel value** at the same position.

- They do not consider the neighborhood or surrounding pixels.
- Operations are performed **per-pixel basis** → fast and simple.
- Mainly used for **brightness adjustment, contrast enhancement, image inversion, thresholding, and segmentation**.

Mathematically:

$$s=T(r)$$

where **r** = input pixel intensity, **s** = output pixel intensity, and **T** = transformation function.

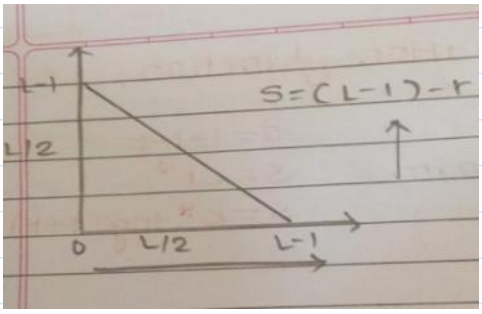
1. Digital Negative(image Inversion)

Digital negative (also called image inversion) is a point processing operation in which each pixel intensity value is subtracted from the maximum possible value (L−1).

- It converts dark areas into light and light areas into dark.
- Used to enhance details in dark regions and in applications like medical or satellite imaging.

Formula:

$$s = (L - 1) - r$$



Applications:

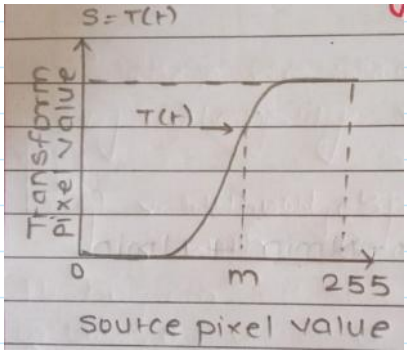
- Medical imaging (X-rays)
- Satellite imaging (terrain details)
- Enhancing dark details

2. Thresholding

Thresholding is a point processing technique used to segment objects from the background.
It converts a grayscale image into a binary image based on a threshold value **T**.

- Pixels with intensity < T → set to black (0).
- Pixels with intensity ≥ T → set to white (255).

$$s(x,y) = \begin{cases} 0 & r(x,y) < T \\ 255 & r(x,y) \geq T \end{cases}$$



Applications: OCR, document scanning, medical imaging, industrial inspection.

3. Intensity Transformations

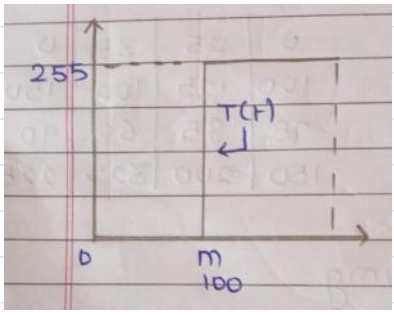
Intensity transformations are point processing techniques where the output pixel value is calculated by applying a mathematical function to the input pixel value.
They are used for **brightness adjustment, contrast enhancement, image correction, and feature highlighting.**
There are 3 main types:

a) Logarithmic Transformation:

This transformation expands low intensity (dark) pixel values while compressing high intensity (bright) pixel values. It improves details in darker regions of an image.

Formula:

$$s = c \cdot \log(1 + r)$$



Where:

- **s** → output pixel
- **r** → input pixel
- **c** → scaling constant

Applications:

- Enhances details in dark regions
- Spectrum display
- Medical and astronomical imaging

b) Power-Law (Gamma) Transformation

This transformation adjusts image brightness and contrast using a power-law curve controlled by the gamma (γ) parameter.

Formula:

$$s = c \cdot r^{\gamma}$$

Where:

- **γ < 1** → enhances brightness (useful for dark images)
- **γ > 1** → reduces brightness (useful for bright images)

Applications:

- Display devices
- photography brightness correction

c) Contrast Stretching (Piecewise Linear Transformation)

Contrast stretching expands the dynamic range of gray levels in an image, making dark areas darker and bright areas brighter for better visibility.

Formula:

$$s = ((r - r_{min}) / (r_{max} - r_{min})) \cdot (L - 1)$$

Where:

- **r_min, r_max** → minimum & maximum intensity values
- **L** → number of gray levels

Applications

- Medical (X-ray, MRI),
- satellite images
- low-contrast photo enhancement

4. Linear vs Non-Linear Transformations

Aspect	Linear Transformation	Non-Linear Transformation
Formula	s = a*r + b	s = c*log(1+r), s = c*r^γ
Nature	Straight line	Curved mapping
Effect	Simple brightness/contrast	Enhances dark/bright separately
Example	Contrast Stretching	Logarithmic, Gamma
Advantages	Easy, fast	Flexible, reveals hidden details
Applications	General correction	Medical, satellite, displays

Steps to Perform

1. Start
2. Import libraries (OpenCV, NumPy, Matplotlib)
3. Read grayscale image
4. Apply Digital Negative, Thresholding, Log, Gamma, Contrast Stretching
5. Display results

6. Stop

Output

Attach screenshots of all transformations: Original, Negative, Thresholding, Log Transform, Gamma Transform, Contrast Stretching.

Conclusion

Thus, the program for implementing point processing techniques such as Digital Negative, Thresholding, Intensity Transformation, and Contrast Stretching using Python and OpenCV was successfully executed and verified.