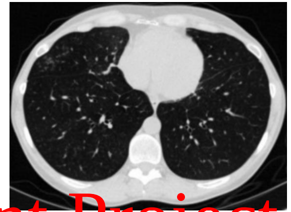


Computer

Image Processing I

Image processing

- **Image processing** = image in > image out

- Aims to suppress relevant information

- Prepares images terpretation

- **Image analysis** = image in > feat

- **Computer vision** = image in > interpretation out

Types of image processing

- Two main types of image processing operations:
  - Spatial domain operations (in image space)
  - Frequency doma[illegible]ier space)

- Two main types of spatial domai[illegible]s:
  - Point operations (intensity transformations on individual pixels)
  - Neighbourhood operations (spatial filtering on groups of pixels)

Types of image processing

- Two main types of image processing operations:

  - Spatial domain operations (in image space)
  - Frequency doma ier space)

- Two main types of spatial domai s:
  - Point operations (intensity transformations on individual pixels)
  - Neighbourhood operations (spatial filtering on groups of pixels)

Topics and learning goals

- Describe the workings of basic point operations

Contrast stretching, thresholding, inversion, log/power transformations

- Understanding a ogram

Histogram specification, equalization

- Defining arithmetic and logical operations

Summation, subtraction, AND, OR, et cetera

Spatial domain operations

- General form of spatial domain operations

$$g(x,y) = T[f(x,y)]$$

where

$f(x,y)$ is the input im

$g(x,y)$ is the processed image

$T[\,\cdot\,]$ is the operator applied at (x,y)

Spatial domain operations

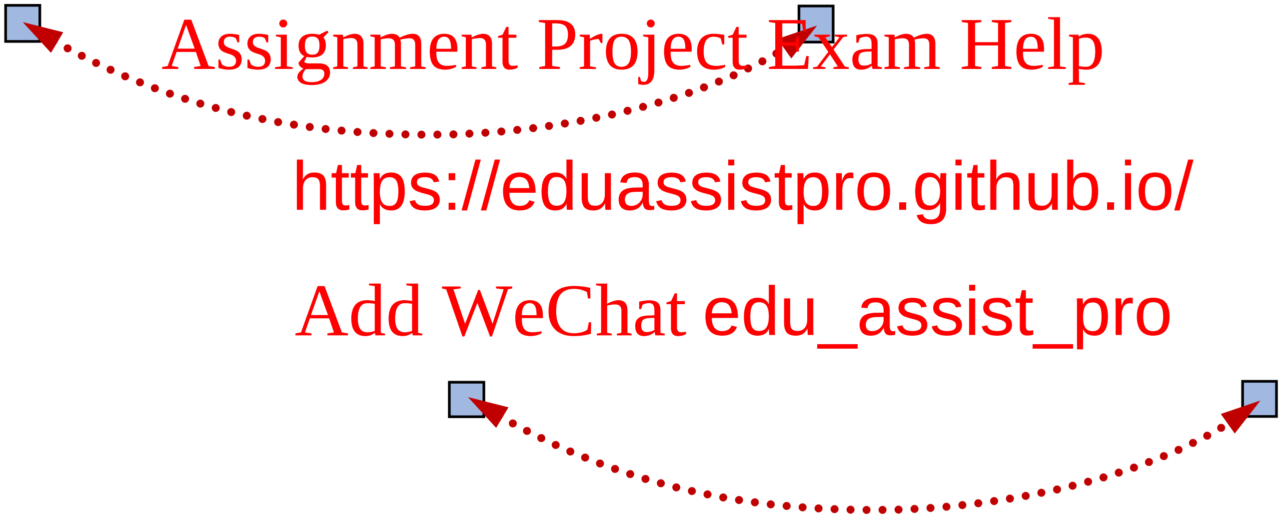
- Point operations: T operates on individual pixels

$$T: \mathbb{R} \longrightarrow \mathbb{R} \qquad g(x, y) = T\big(f(x, y)\big)$$

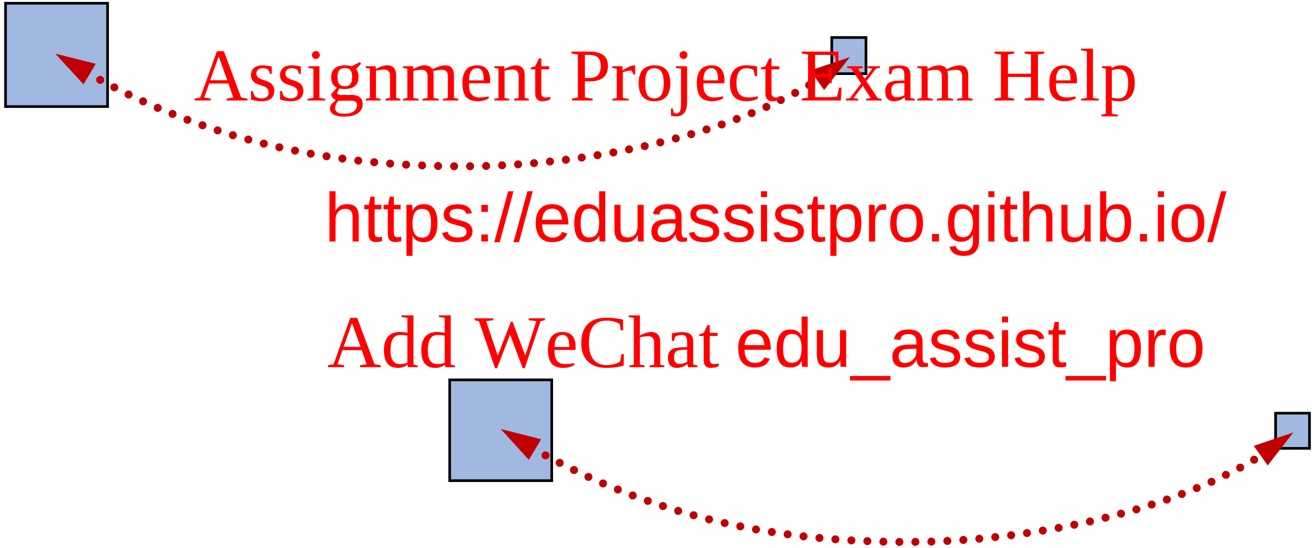
- Neighbourhood operations: T op ultiple pixels

$$T: \mathbb{R}^2 \longrightarrow \mathbb{R} \quad g(x, y) = T(f(x, y), f(x+1, y), f(x-1, y), \dots)$$

Point operations

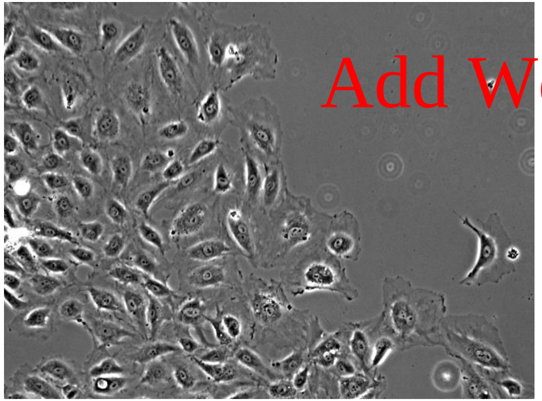


Neighbourhood operations

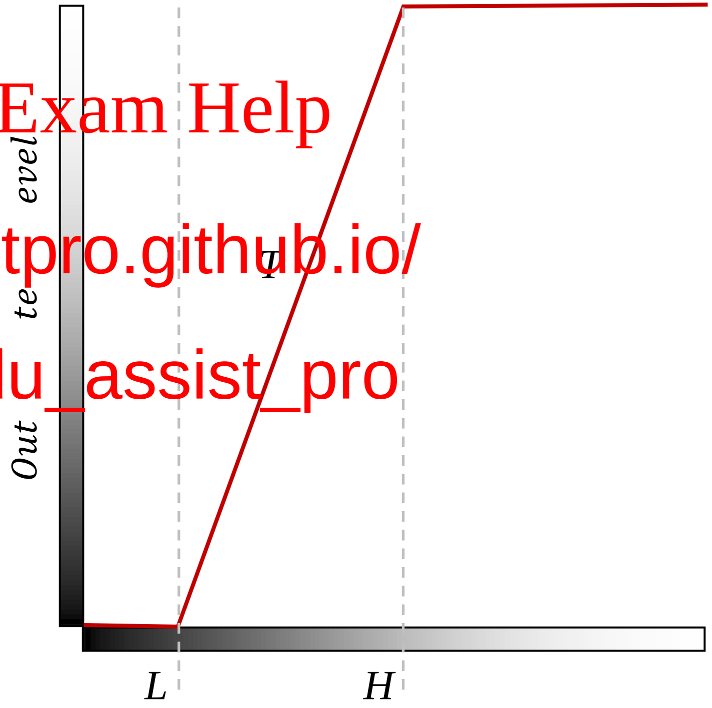


Contrast stretching

Input



Output



Contrast stretching

- Produces images of higher contrast

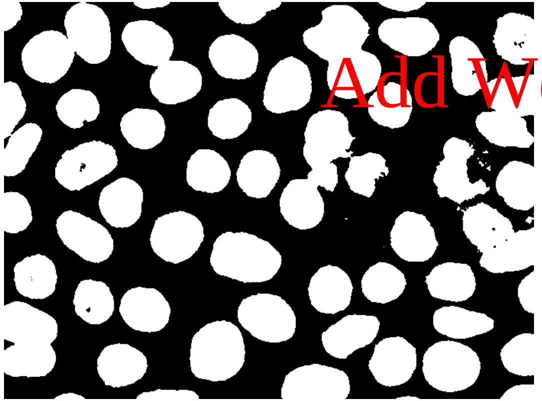
  - Puts values below L in the input to black in the output

  - Puts values abov he output

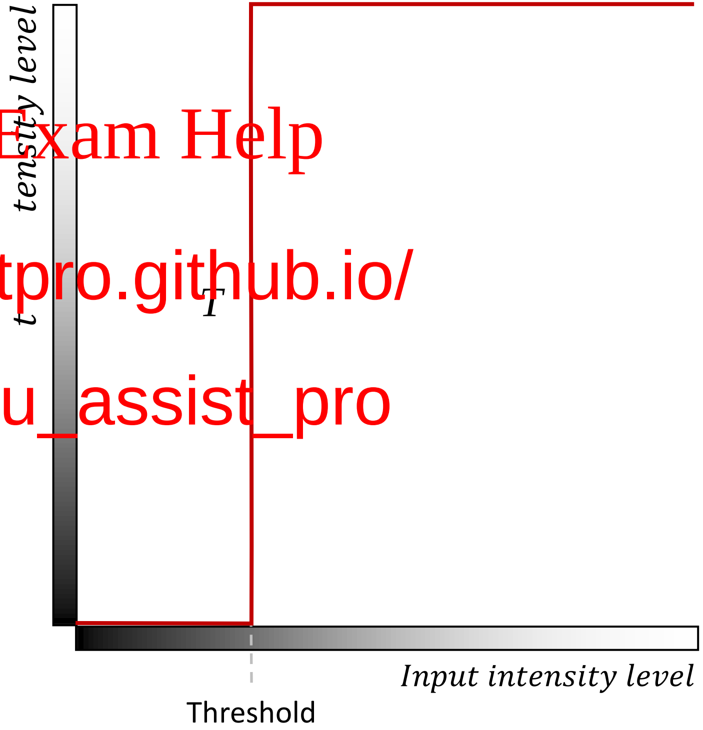
- Linearly scales values between L e input to the maximum range in the output

Intensity thresholding

Input



Output



Intensity thresholding

- Limiting case of contrast stretching
- Produces binary images of gray-scale images
  - Puts values below put
  - Puts values equal/above the threshold output
- Popular method for image segmentation (discussed later)
- Useful only if object and background intensities are very different

Automatic intensity thresholding

- Otsu’s method for computing the threshold

Exhaustively searches for the threshold minimising the intra-class variance

Equivalent to maxi … uch faster to compute)

$$\sigma_B^2 = p_0 p_1 (\mu_0 \ldots)$$

Here, p_0 is the fraction of pixels below the threshold (class 0), p_1 is the fraction of pixels equal to or above the threshold (class 1), μ_0 and μ_1 are the mean intensities of pixels in class 0 and class 1, σ_0^2 and σ_1^2 are the intensity variances, and $p_0 + p_1 = 1$ and $\sigma_0^2 + \sigma_1^2 = \sigma^2$

Otsu thresholding

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Automatic intensity thresholding

- Iso-data method for computing the threshold

1. Select an arbitrary initial threshold t
2. Compute μ_0 and ld
3. Update the threshold to the mean $= (\mu_0 + \mu_1)/2$
4. If the threshold changed in Step 3,

Upon convergence, the threshold is midway between the two class means

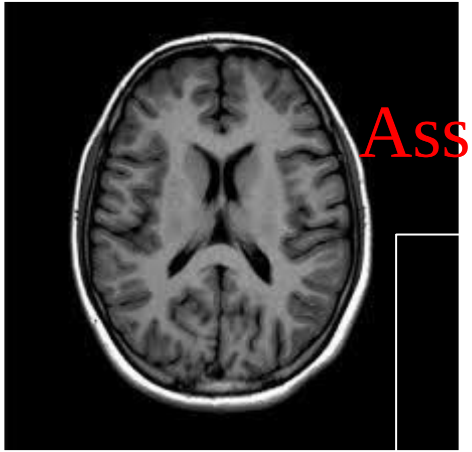
Iso-data thresholding

Assignment Project Exam Help

https://eduassistpro.github.io/

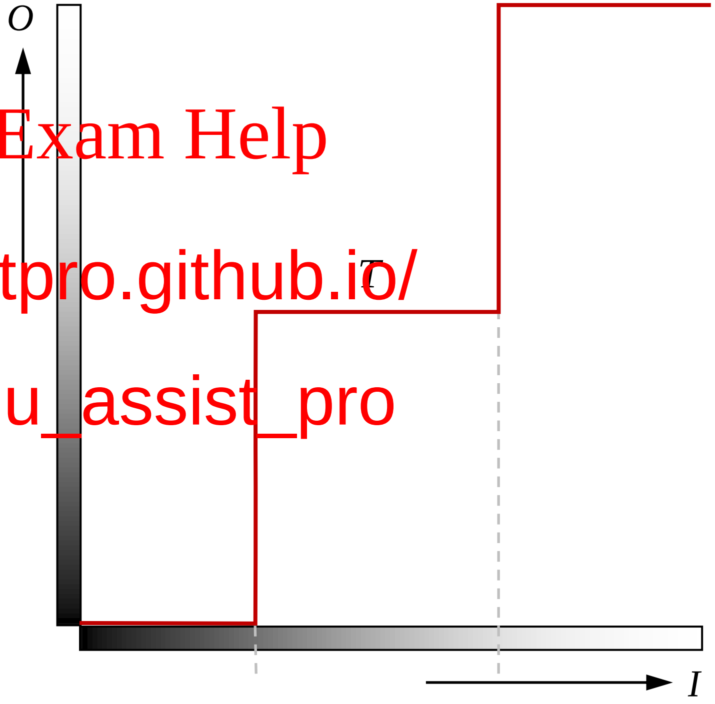
Add WeChat edu_assist_pro

Multi-level thresholding



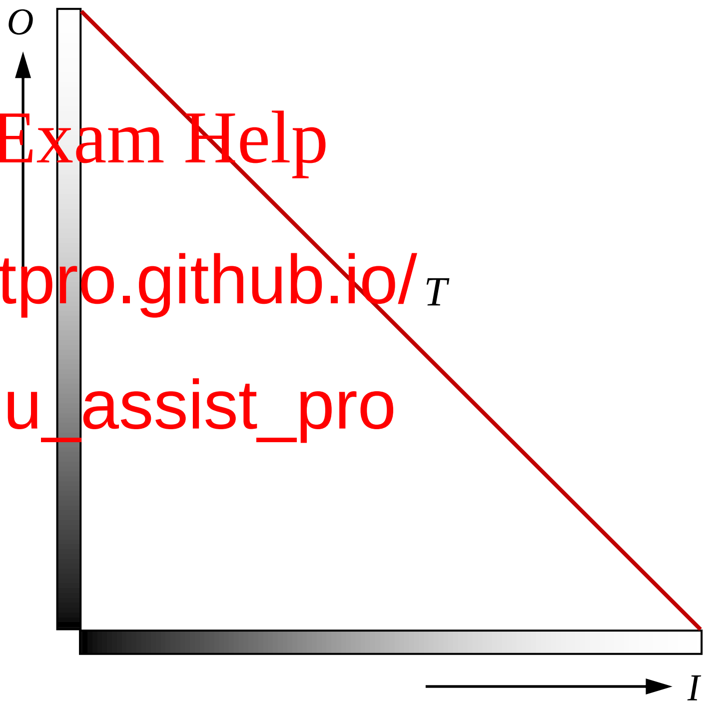
Input

Output



Intensity inversion

Input



Intensity inversion

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Useful for enhancing gray/white details in images within dominant black areas

Log transformation

- Definition of log transformation

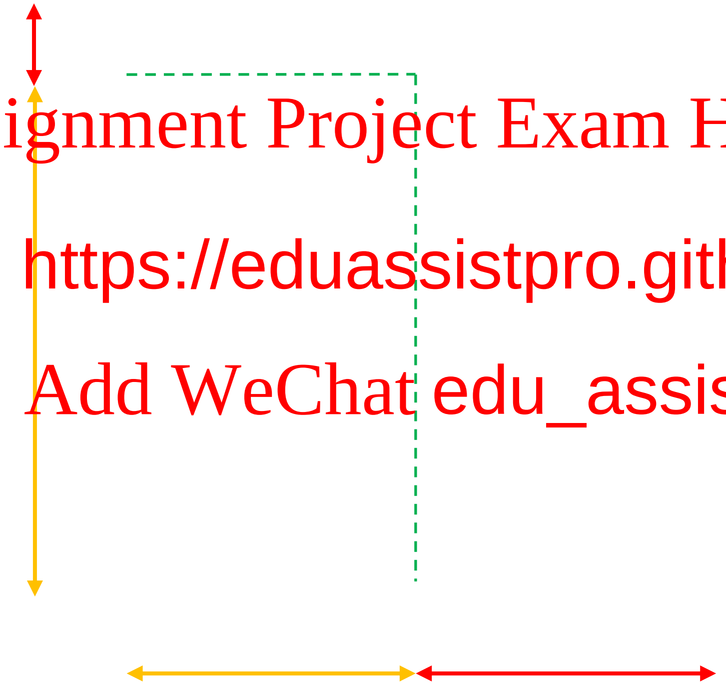
$$s = c \log(1 + r)$$

where r is the input … sity, and c is a constant

– Maps narrow range of low gray-level … ider range of output values, and opposite for higher gray-

– Also compresses dynamic range of images with large variations in pixel values (such as Fourier spectra, discussed later)

Log transformation



Power transformation

- Definition of power transformation

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

where c and γ are c

  - Similar to log transformation

  - Represents a whole family of transf arying γ
  - Many devices respond according to a power law (gamma correction)
  - Useful for general-purpose contrast manipulation

Power transformation

Power transformation

Input

$\gamma = 3$

$c = 1$

$\gamma = 4$

$\gamma = 5$

Piecewise linear transformations

- Complementary to other transformation methods

- Enable more fine-tuned design of transformations

- Can have very co

- Requires more user input

Piecewise contrast stretching

- One of the simplest piecewise linear transformations
- Increases the dynamic range of gray levels in images
- Used in display d to span full range

Piecewise contrast stretching

Transform

Input

Transformed

Binary Thresholding

Gray-level slicing

- Used to highlight specific range of gray levels

- Two different slicing approaches:

1) High value for a [illegible] rest and low value for all others (produces a binary image)

2) Brighten a desired range of gray levels while preserving background and other gray-scale tones of the image

Gray-level slicing

Transform 1

Transform 2

Input

Result of Transform 1

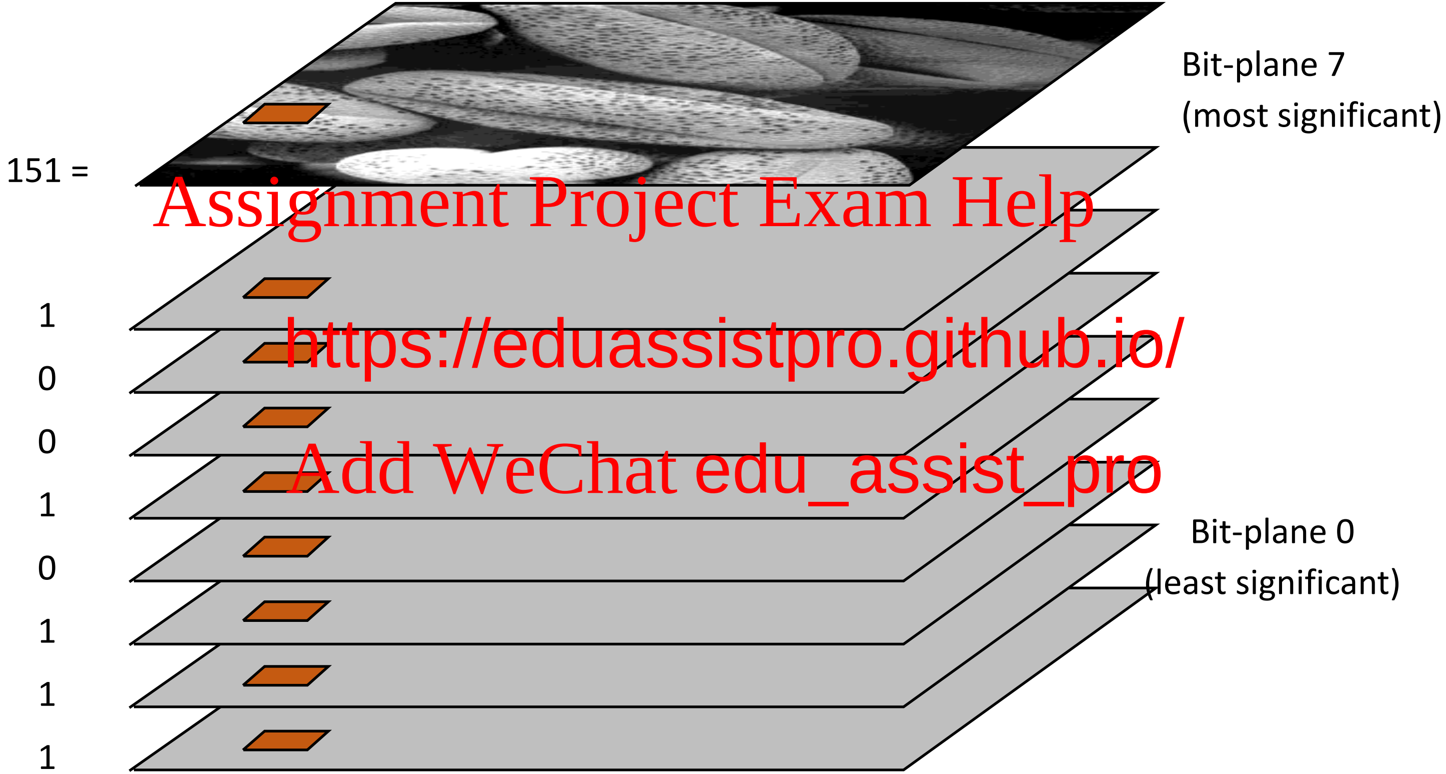
Bit-plane slicing

- Highlights contribution to total image by specific bits

- An image with *n*-bits/pixel has *n* bit-planes

- Slicing can be us n

Bit-planes of an 8-bit image



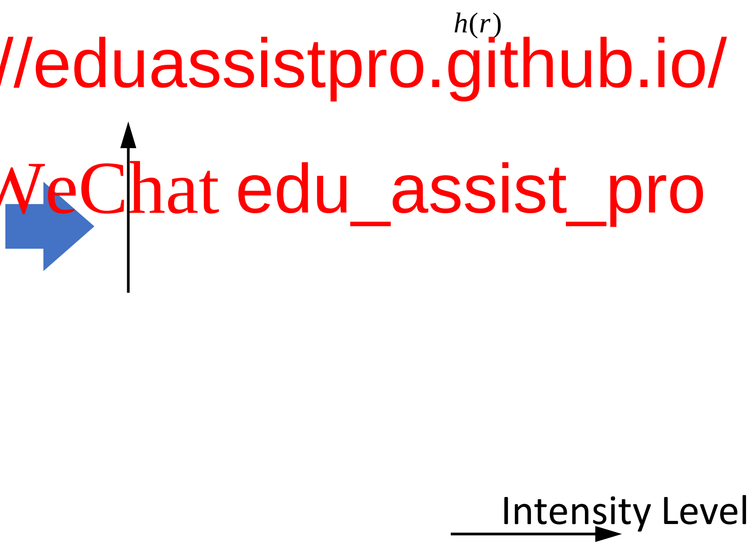
Bit-planes of an 8-bit image

Input

Bit-planes

Histogram of pixel intensities

- For every possible intensity level, count the number of pixels having that level, and plot the pixel counts as a function of the level



8-bit image

$L = 2^8 = 256$

N = #pixels

$$\sum_{r=0}^{L-1} h(r) = N$$

Normalized histogram = probability function

$$\frac{1}{N} h(r) = p(r)$$

Histogram processing

- **Histogram equalization**

Aim: To get an image with equally distributed intensity levels

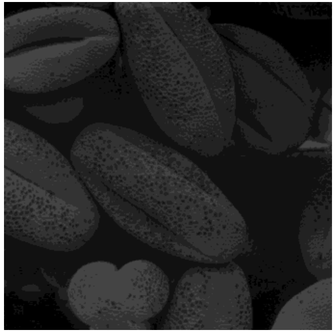
over the full inten

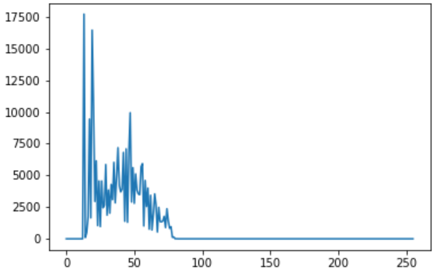
- **Histogram specification** (also called **atching**)

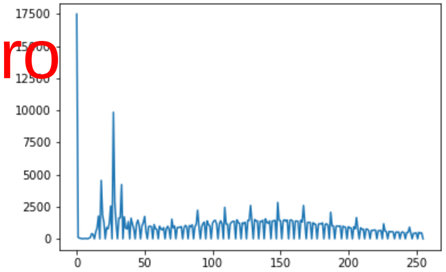
Aim: To get an image with a specifi istribution,

determined by the shape of the histogram

Histogram processing







Histogram equalization

Enhances contrast for intensity values near histogram maxima an decreases contrast ne histogram minima

ogram bins are much more “equal” here

37

Histogram equalization

- Let $r \in [0, L-1]$ represent gray levels of the image

  $r = 0$ represents black and $r = L - 1$ represents white

- Consider transform $(\ )$, satisfying

1) $T(r)$ is single-valued and monotonicall $\leq r \leq L-1$

   This guarantees that the inverse transf

2) $0 \leq T(r) \leq L-1$ for $0 \leq r \leq L-1$

   This guarantees that the input and output ranges will be the same

Histogram equalization (continuous case)

Consider r and s as continuous random variables over $[0, L-1]$ with PDFs $p_r(r)$ and $p_s(s)$

If $p_r(r)$ and $T(r)$ are known and $T^{-1}(s)$ satisfies monotonicity, then, from probability theory

$$p_s(s) = p_r(r)\left|\frac{dr}{\quad}\right|$$

Let us choose: $s = T(r) = (L$ $()$

This is the CDF of r which satisfies conditions (1) and (2)

Now: $\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(\xi)d\xi\right] = (L$

Therefore: $p_{s(s)} = p_{r(r)}\left|\frac{1}{(L-1)p_r(r)}\right| = \frac{1}{L-1}$ for $0 \leq s \leq L-1$

This is a uniform distribution!

Histogram equalization (discrete case)

For discrete values we get probabilities and summations instead of PDFs and integrals:

$$p_r(r_k) = n_k/MN \quad \text{for} \quad k = 0, 1, \ldots, L-1$$

where MN is total number o ixels with gray level r_k

and L is the total number of

Thus $s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k}$ $, L-1$

This transformation is called *histogram equalization*

However, in practice, getting a perfectly uniform distribution for discrete images is rare

Histogram matching (continuous case)

Assume that r and s are continuous intensities and $p_z(z)$ is the target distribution for the output image

From our previous analysis we know that the following transformation results in a uniform distribution:

$$(\ ) \qquad r \qquad (\ )$$

Now we can define a function G

$$G(z) = (L-1)\int_0^z p_z(\ )$$

Therefore:

$$z = G^{-1}(s) = G^{-1}[T(r)]$$

Histogram matching (discrete case)

For discrete image values we can write:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j$$

and

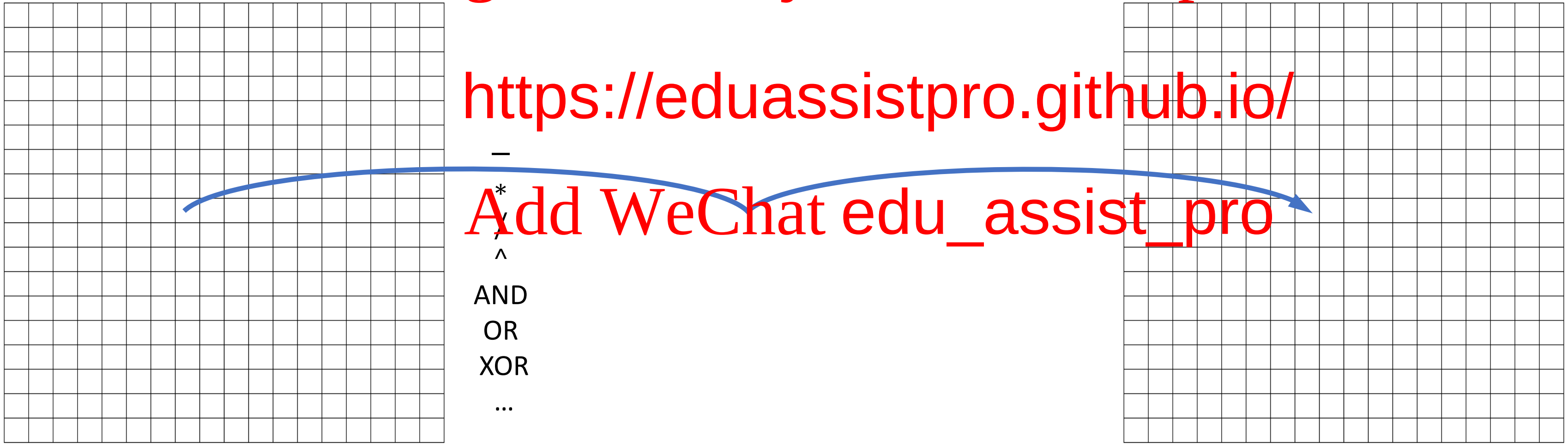
$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)$$

therefore:

$$z_q = G^{-1}(s_k)$$

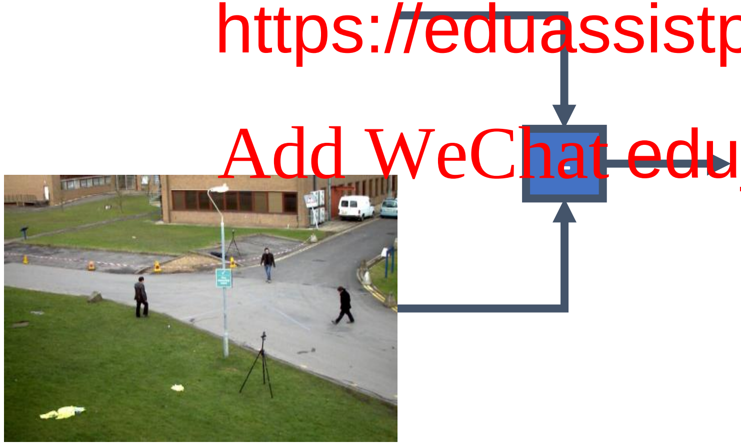
Arithmetic and logical operations

- Defined on a pixel-by-pixel basis between two images



Arithmetic and logical operations

- Useful arithmetic operations include addition and subtraction



Arithmetic and logical operations

- Useful logical operations include AND and OR

Input Mask Input AND Mask

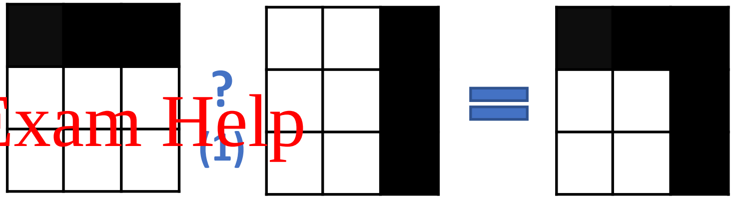
Arithmetic/Logic Operations

- on pixel-by-pixel basis between 2 or more images

- AND and OR o *king*- selecting subimages as

- subtraction and addition are the arithmetic operations

Chapter 3
Image Enhancement in the
Spatial Domain



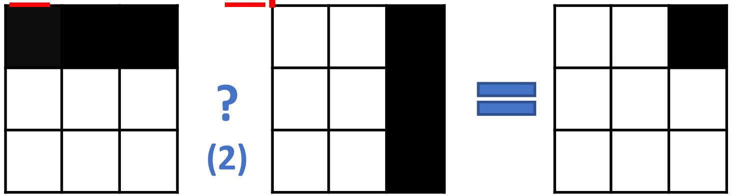


Image Averaging

- Noisy image *g(x, y)* formed by adding noise *n(x, y)* to uncorrupted image *f(x, y)*:

$$g(x, y) = f(x, y) + n(x, y)$$

- Assume that at each *(x, y)*, the noise is uncorrelated and has zero average value.

- **Aim:** To obtain sm oisy images

$g_i(x, y), \; i =$

$$g(x, y) \approx \frac{1}{K}\sum_{i=1}^{K} g_i(x, y)$$

- As K increases, the variability of the pixel values decreases

- assumes that images are spatially registered

Chapter 3
Image Enhancement in the Spatial Domain

a	b
c	d
e	f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64,$ and 128 noisy images. (Original image courtesy of NASA.)

Spatial Filtering

- These methods use a small *neighbourhood* of a pixel in the input image to produce a new brightness value for that pixel
- Also called *filtering* techniques

- Neighbourhood of ... tangular subimage centred at (x, y) - c ... *late / window*

- A *linear transform* ... utput image *g(i, j)* as a linear combination of brightnesses in ... urhood of the pixel in the input image *f(i, j)*, weighted by c

$$g(x, y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} h(i, j) f(x - i, y - j)$$

- This is called a *discrete convolution* with a convolution mask h

Spatial Filtering

Convolution

Smoothing Spatial Filters

Used for blurring, noise reduction

Neighbourhood Averaging (Mean Filter)

( ) =)

- Replace intensity at pixel (x, **rage** of the intensities in a neighbourhoo

- We can also use a **weighted average**, giving more importance to some pixels over others in the neighbourhood- reduces blurring
- Neighbourhood averaging blurs edges

Chapter 3
Image Enhancement in the Spatial Domain

Chapter 3
Image Enhancement in the Spatial Domain

Another example

Consider an image of constant intensity, with widely isolated pixels with different intensity from the background. We wish to detect these pixels.

Use the followin

-1 -1 -1

Smoothing Spatial Filters

- **Aim:** To suppress noise, other small fluctuations in image- may be result of sampling, quantization, transmission, environment disturbances during acquisition
- Uses redundancy in the image data
- May blur sharp edges, so care is needed

Gaussian Filter

$$g(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\left[\frac{x^2+y^2}{2\sigma^2}\right]}$$

- Replace i eighted **average** of the intensities in od of (x, y).

- It is a set of weights that app ofile of a Gaussian function.
- It is very effective in reducing noise and also reducing details (image blurring)

Gaussian Filter

Non-linear Spatial Filters

Also called order-statistics filters- response based on ordering the pixels in the neighbourhood, and replacing centre pixel with the ranking result.

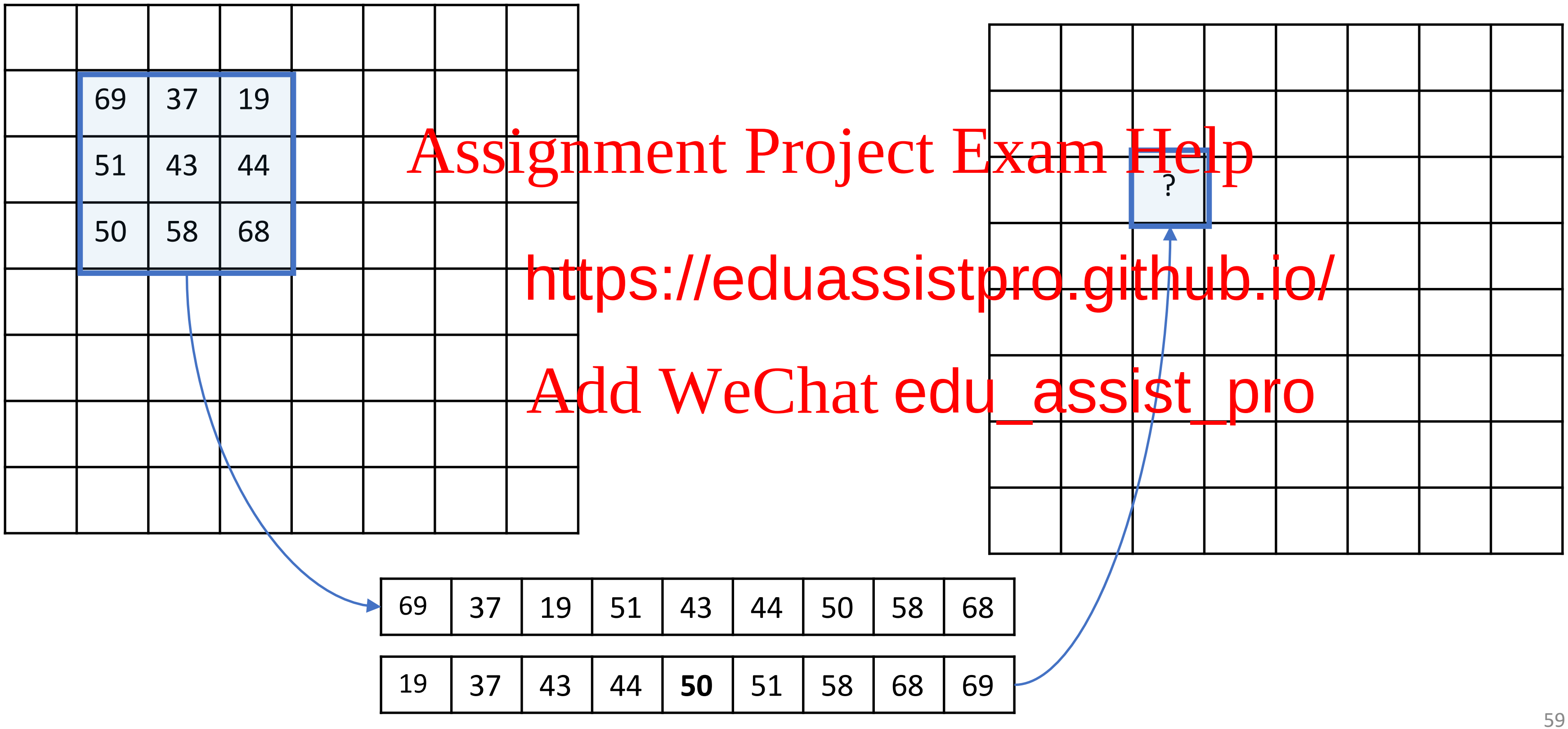
Median Filter

- intensity of each pixel nsities in neighbourhood of that pixel

- Median M of a set of alf the values in the set are less than M and the other half greater than M
- Median filtering forces points with distinct int e like their neighbours, thus eliminating isolated intensity spikes

- Also, isolated pixel clusters (light or dark), whose area is <= n^2/2, are eliminated by nxn median filter
- Good for impulse noise (salt-and-pepper noise)
- Other examples of order-statistics filters are max and min filters

Median Filter



Chapter 3
Image Enhancement in the Spatial Domain

Pooling

Max / average/ median pooling

- Provides translation invariance
- Reduces computations
- Popular in deep c arning)

Sharpening Spatial Filters-Edge Detection

- Goal is to highlight fine details or enhance details that have been blurred
- Spatial differentiat nse of derivative operator is propor of the image at the point where operator is applied
- Image differentiation enhances edge asizes slowly varying gray-level values.

Derivative definitions

- For 1-D function f(x), the first order derivative is approximated as:

$$\frac{df}{dx} =$$

- The second-order derivative is approxi

$$\frac{d^2 f}{dx^2} = f\,(x+1) + f\,(x-1) - 2\,f\,(x)$$

- These are partial derivatives, so that extension to 2D is easy.

Chapter 3
Image Enhancement in the Spatial Domain

Basic idea - Derivatives

- Horizontal scan of the image
- Edge modelled as a ramp- to represent blurring due to sampling
- First derivative is
  - Non-zero along ramp
  - zero in regions of constan
  - constant during an intensit
- Second derivative is
  - Nonzero at onset and end of ramp
  - Stronger response at isolated noise point
  - zero everywhere except at onset and termination of intensity transition
- Thus, magnitude of first derivative can be used to detect the presence of an edge, and sign of second derivative to determine whether a pixel lies on dark or light side of an edge.

a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Summary - Derivatives

- First-order derivatives produce thicker edges, have stronger response to gray-level step
- Second-order de sponse to fine detail (thin lines, isolat sponse at step changes in gray level

Gradient Operator

First-order derivatives implemented using magnitude of the gradient

For function *f(x, y)*, the gradient of *f* at *(x, y)* is **G** with *x* and *y* components G_x , G_y

The magnitude of the

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2}$$

This is commonly approximated by: $G[f(x, y$

G_x and G_y are linear and may be obtained by

We use numerical techniques to compute these- give rise to different masks, e.g. Roberts' 2x2 cross-gradient operators, Sobel's 3x3 masks

Chapter 3
Image Enhancement in the Spatial Domain

The Laplacian

Second order derivatives based on the Laplacian.
For a function *f (x, y),* the Laplacian is defined by

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

This is a linear operator, a

In discrete form:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

and similarly in *y* direction.

Summing them gives us

$$\Delta^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Chapter 3
Image Enhancement in the Spatial Domain

Laplacian ctd

- There are other forms of the Laplacian- can include diagonal directions, for example

- Laplacian ities and produces

- The background can be reco g or subtracting the Laplacian image to the original image

Chapter 3
Image Enhancement in the Spatial Domain

Chapter 3
Image Enhancement in the Spatial Domain

Chapter 3
Image Enhancement in the
Spatial Domain

Chapter 3
Image Enhancement in the
Spatial Domain

Padding

- When we use spatial filters for pixels on the boundary of an image, we do not have enough neighbours

- To get an image mage

  - ***Zero***: set all pixels
  - ***Constant***: set all pixels outside the source i border value
  - ***Clamp***: repeat edge pixels indefinitely
  - ***Wrap***: copy pixels from opposite side of the
  - ***Mirror***: reflect pixels across the image edge

Padding Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Szeliski, “Computer Vision”, Chapter 3

References and acknowledgements

- Chapter 3 of Gonzalez and Woods 2002
- Sections 3.1-3.3 of Szeliski
- Some images dr [illegible] s