Image-Processing Application

Learning Outcomes

At the end of this module, you will be able to:

- Outline the operation and types of edge detection algorithms.
- Describe the operation of grayscaling and mathematical derivation of gradient magnitude.
- Compare and contrast the Sobel kernels with other existing kernels including Roberts Cross and Prewitt.

Edge Detection

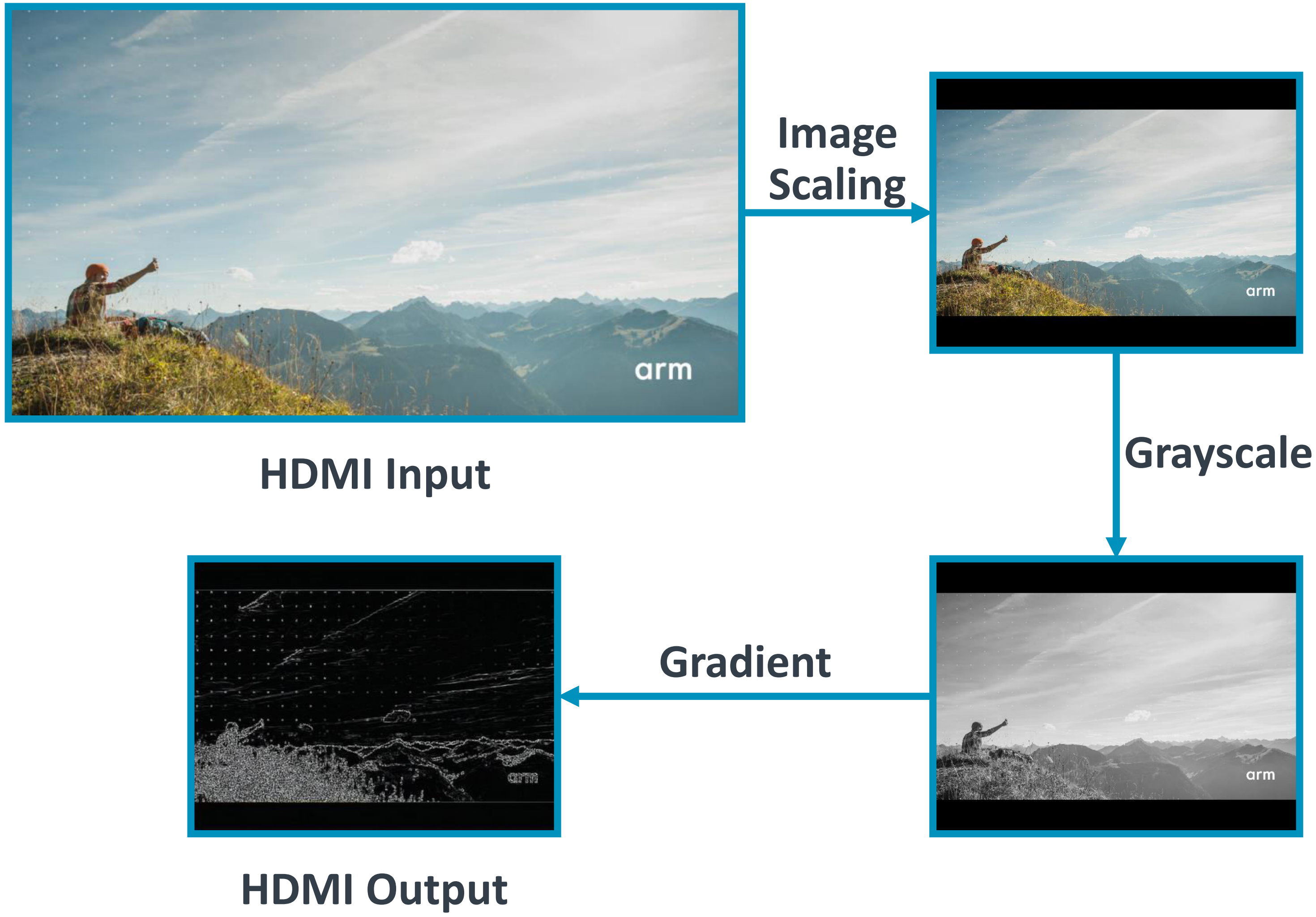
- Edge detection is a basic mathematical operation widely used in image processing and computer vision.
- It detects sharp changes in image brightness.

Picture author: JonMcLoone

Edge Detection

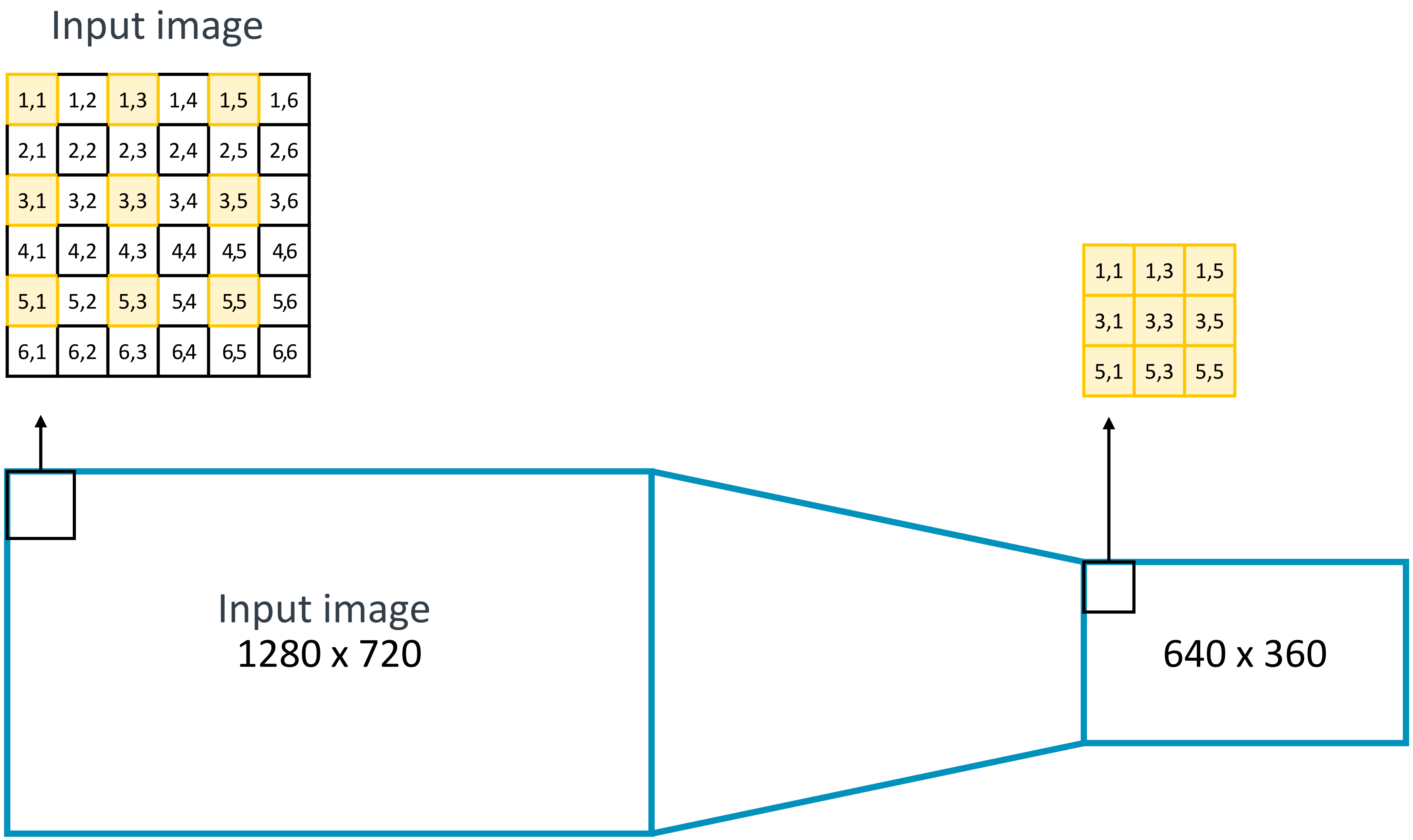
- Most edge detection algorithms can be broadly grouped into two categories:
  - Search-based edge detection
  - Zero-crossing based edge detection
- Search-based methods
  - Detect edges by searching for local directional maxima of edge strength
  - Usually a first-order derivative expression, such as the gradient magnitude
- Zero-crossing based methods
  - Search for zero crossings in a second-order derivative expression computed from the image

Edge Detection



arm

Image Scaling

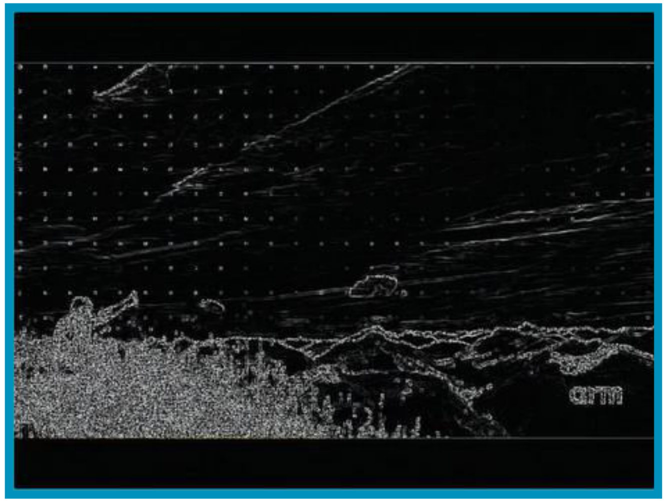


Grayscale

Grayscale

- $A = 0.299R + 0.587G + 0.114B$
  - ——PAL/NTSC
- $A = 0.2126R + 0.7152G + 0.0722B$
  - ——HDTV/sRGB
- $A = \frac{19595}{65536}R + \frac{38470}{65536}G + \frac{7471}{65536}B$
  - ——simplified equation we can use

Intensity Gradient Magnitude



Intensity Gradient Magnitude

- The gradient magnitude of an image can be computed as
- $|G| = \sqrt{G_x^2 + G_y^2}$
    - where G_x and G_y are estimates of first-order image derivatives.
- Different gradient operators can be applied to estimate first-order derivatives from the input image or a smoothed version of it. The simplest approach is to use central differences:
- $G_x(x,y) = -1/2 \cdot A(x-1,y) + 0 \cdot A(x,y) + 1/2 \cdot A(x+1,y)$
  $G_y(x,y) = -1/2 \cdot A(x,y-1) + 0 \cdot A(x,y) + 1/2 \cdot A(x,y+1)$
- or write it as a convolution operator: $G_x = [-1/2 \quad 0 \quad 1/2] * A,\qquad G_y = \begin{bmatrix} 1/2 \\ 0 \\ -1/2 \end{bmatrix} * A$
- which requires intensity values of four pixels around the pixel considered the central (right, left, up, and down).

Intensity Gradient Magnitude

- Other convolution operators:
  - Sobel

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A, \qquad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

These kernels are designed to respond maximally to horizontal and vertical edges (relative to the pixel grid), with one kernel for each of the two perpendicular orientations.

  - Roberts Cross

$$G_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} * A, \qquad G_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} * A$$

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, with one kernel for each of the two perpendicular orientations.

  - Prewitt

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A, \qquad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A$$

NOTE: The differential masks act as high-pass filters that tend to amplify image noise. Pre-smoothing the image is often needed to reduce effects of the noise. Usually smoothing is done with a low-pass filter.

Software Programming: Edge Detection Algorithm

- Pixels of a frame are stored continuously in DDR memory from the start address.

$B_{1,1}$	$G_{1,1}$	$R_{1,1}$	$B_{1,2}$	$G_{1,2}$	$R_{1,2}$	…	$B_{1,640}$	$G_{1,640}$	$R_{1,640}$	$B_{2,1}$	$G_{2,1}$	$R_{2,1}$	$B_{2,2}$	$G_{2,2}$	$R_{2,2}$	…	$B_{480,640}$	$G_{480,640}$	$R_{480,640}$

- Compute intensity of each pixel:

$$A_{i,j} = 0.30 \cdot R_{i,j} + 0.59 \cdot G_{i,j} + 0.11 \cdot B_{i,j}.$$

- First-order gradient of intensity (Prewitt operator):

- $G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A, G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A$
  where * here is the two-dimensional convolution operation.

- The final pixel's value equals:

$$G = \sqrt{G_x^2 + G_y^2}.$$