AASD 4004
Machine Learning - II

Applied AI Solutions Developer Program

Module 12
Edges
Contours
Shapes

Vejey Gandyer

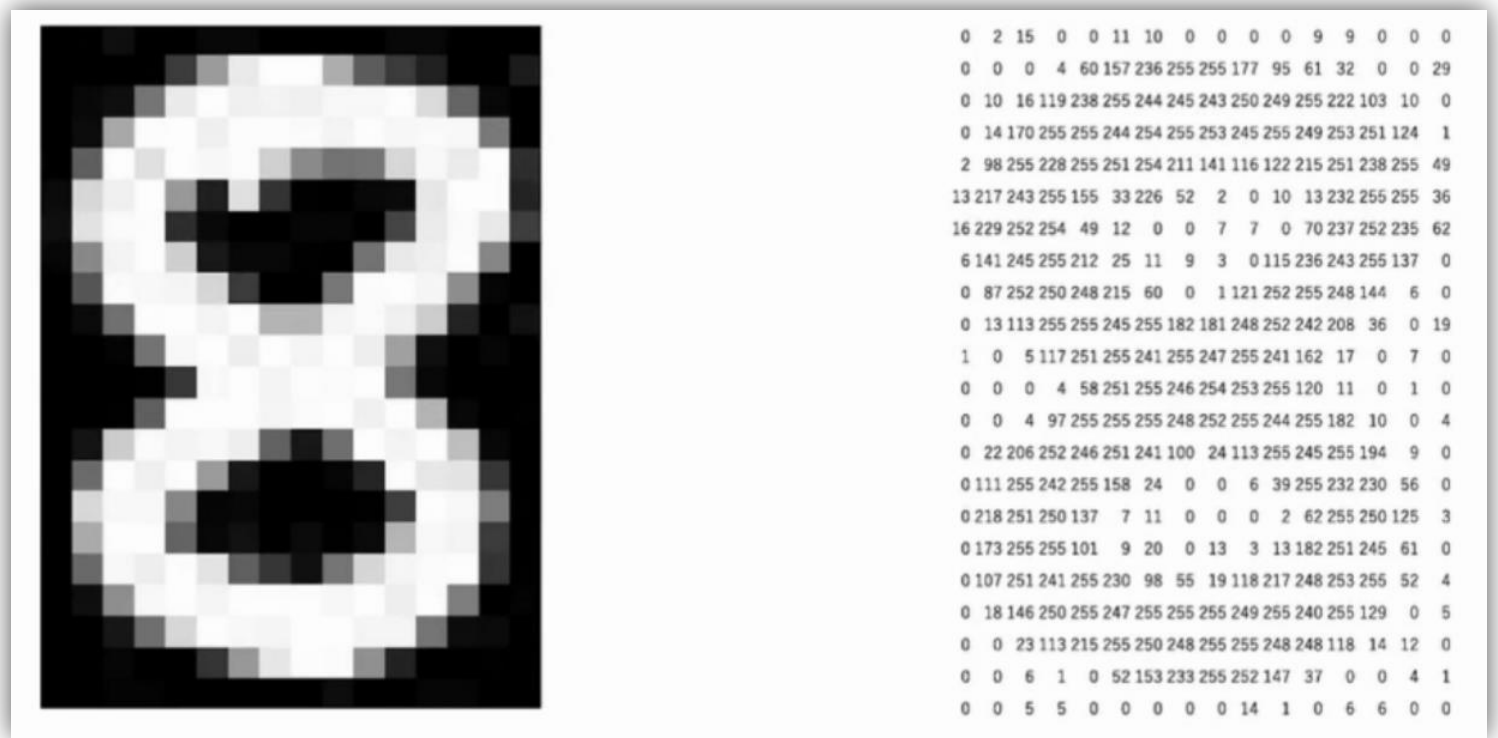
Agenda

- Edge
- Laplacian
- Sobel
- Canny
- Contours
- Watershed
- Shapes

Edges in Images

What is it?

Edges

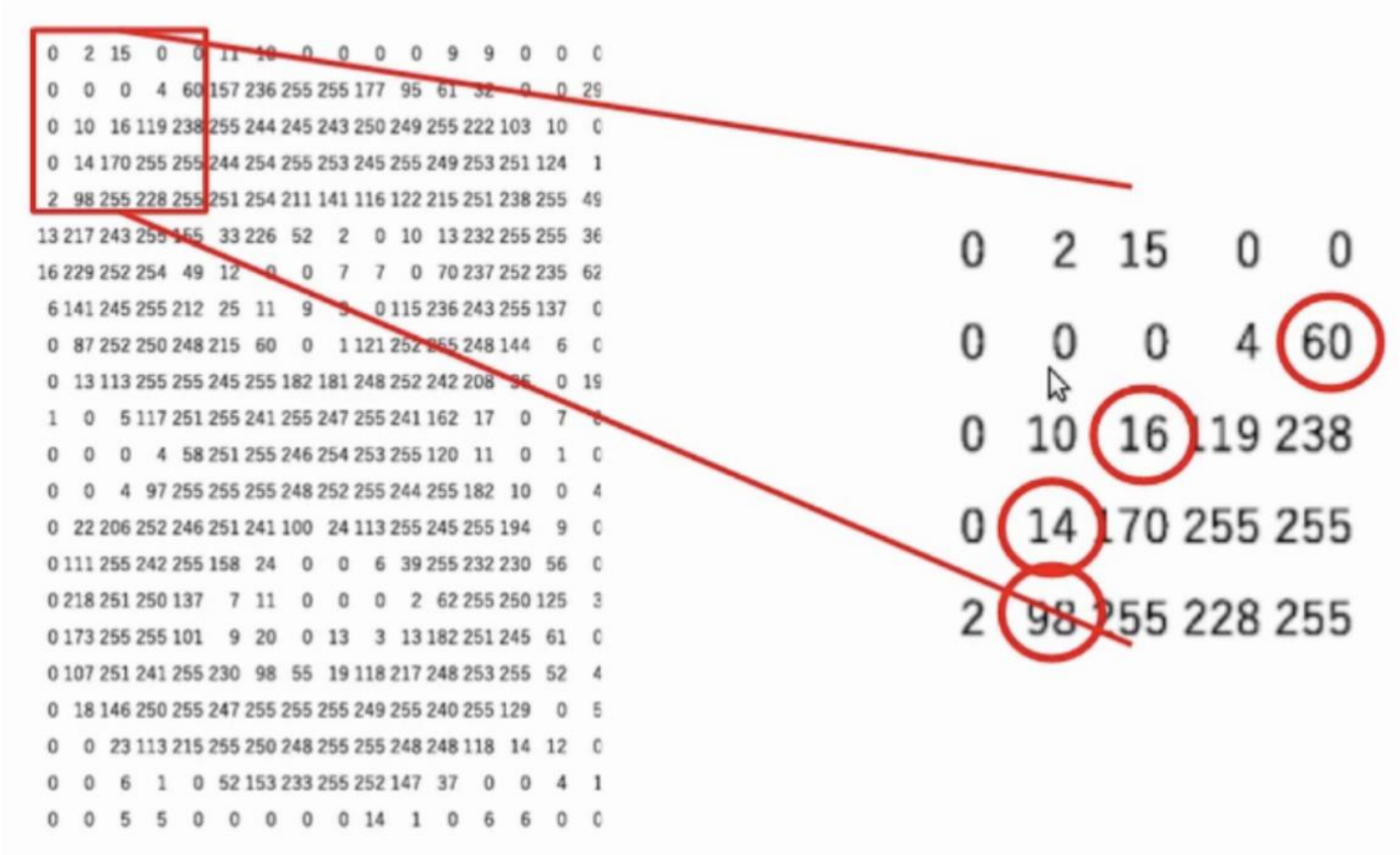


0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

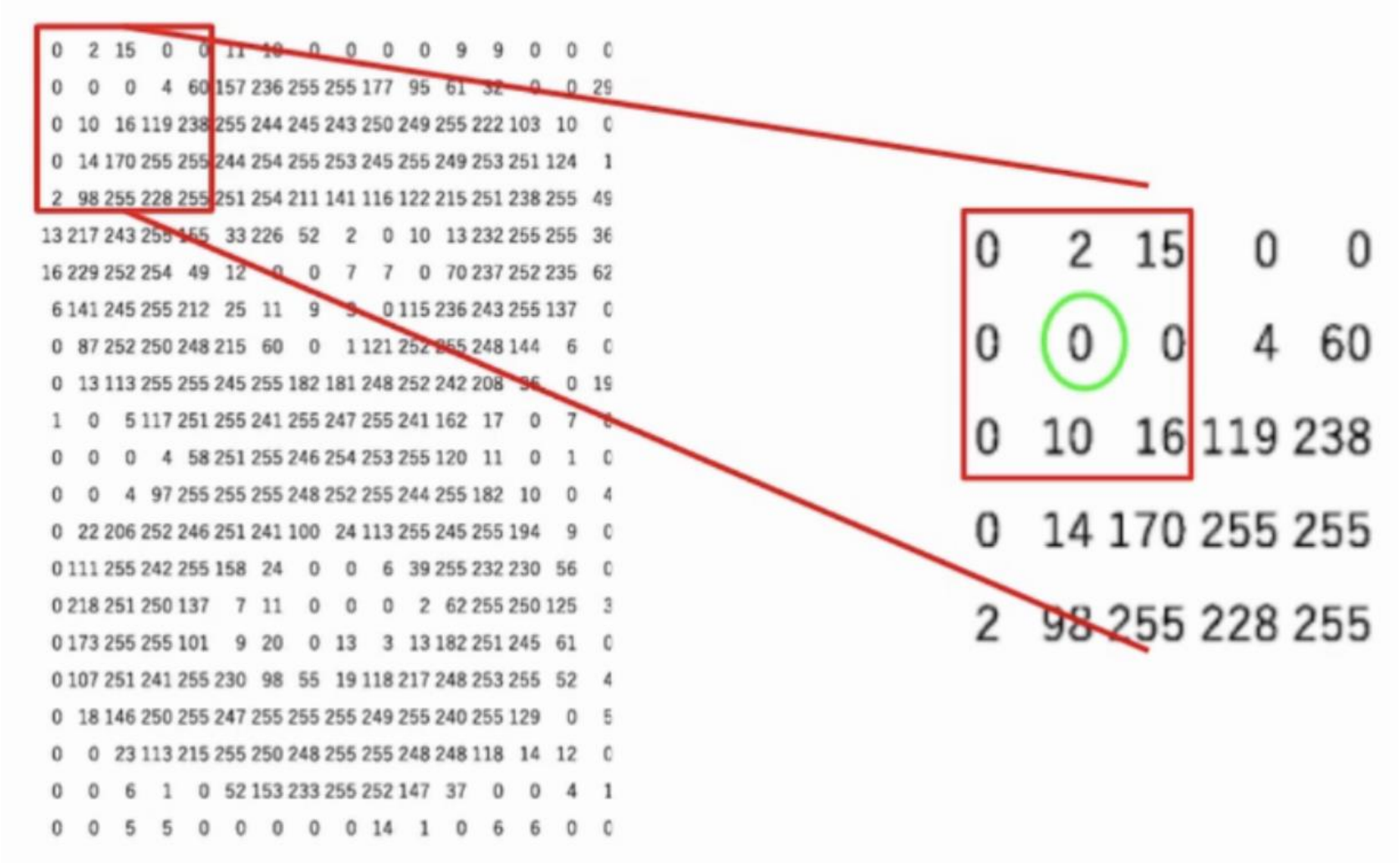
Edge detection is the process of finding boundaries of objects in an image

Edges = Points where brightness of pixel intensities changes **drastically**

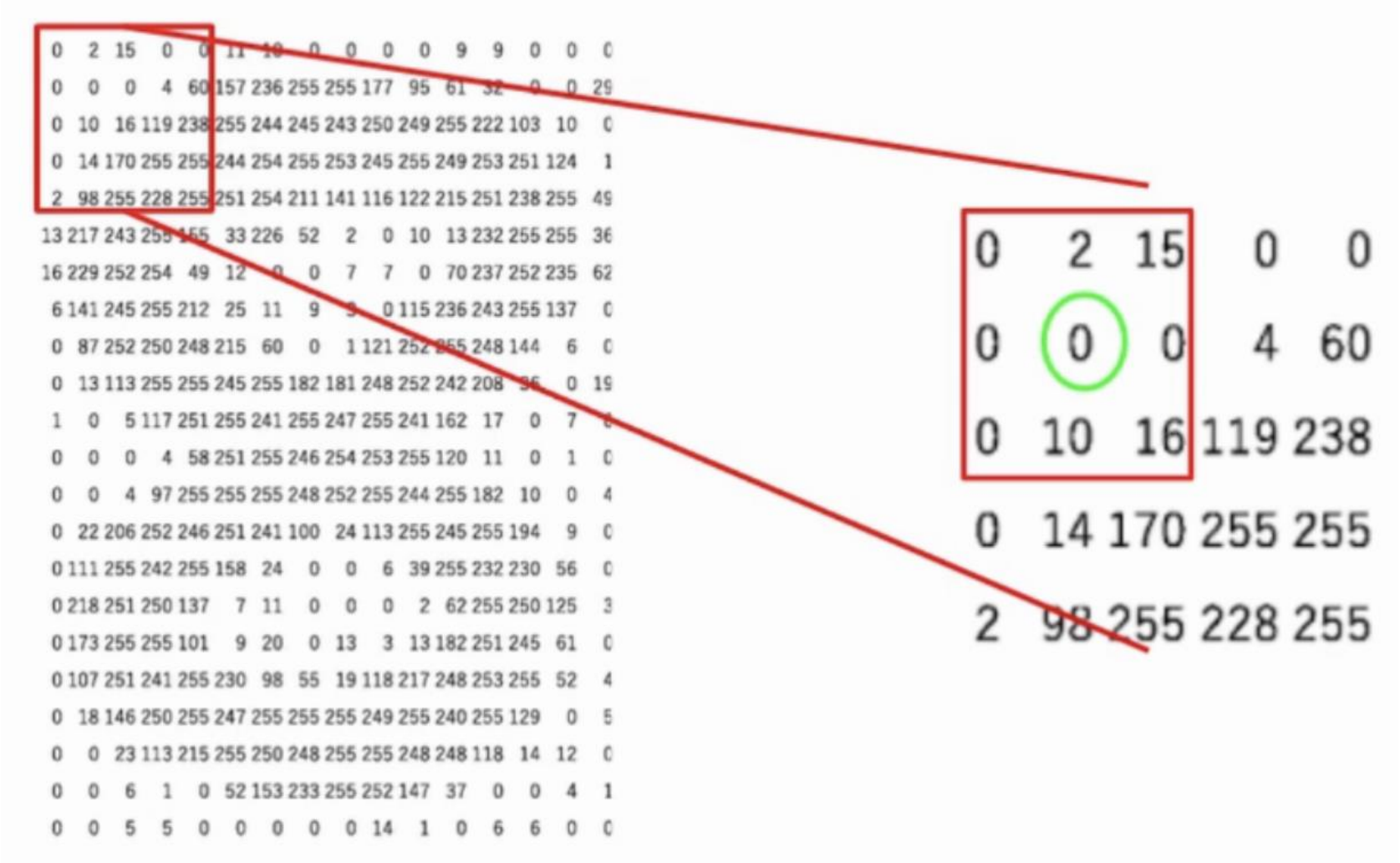
Edge or No edge



Edge or No edge



Edge or No edge

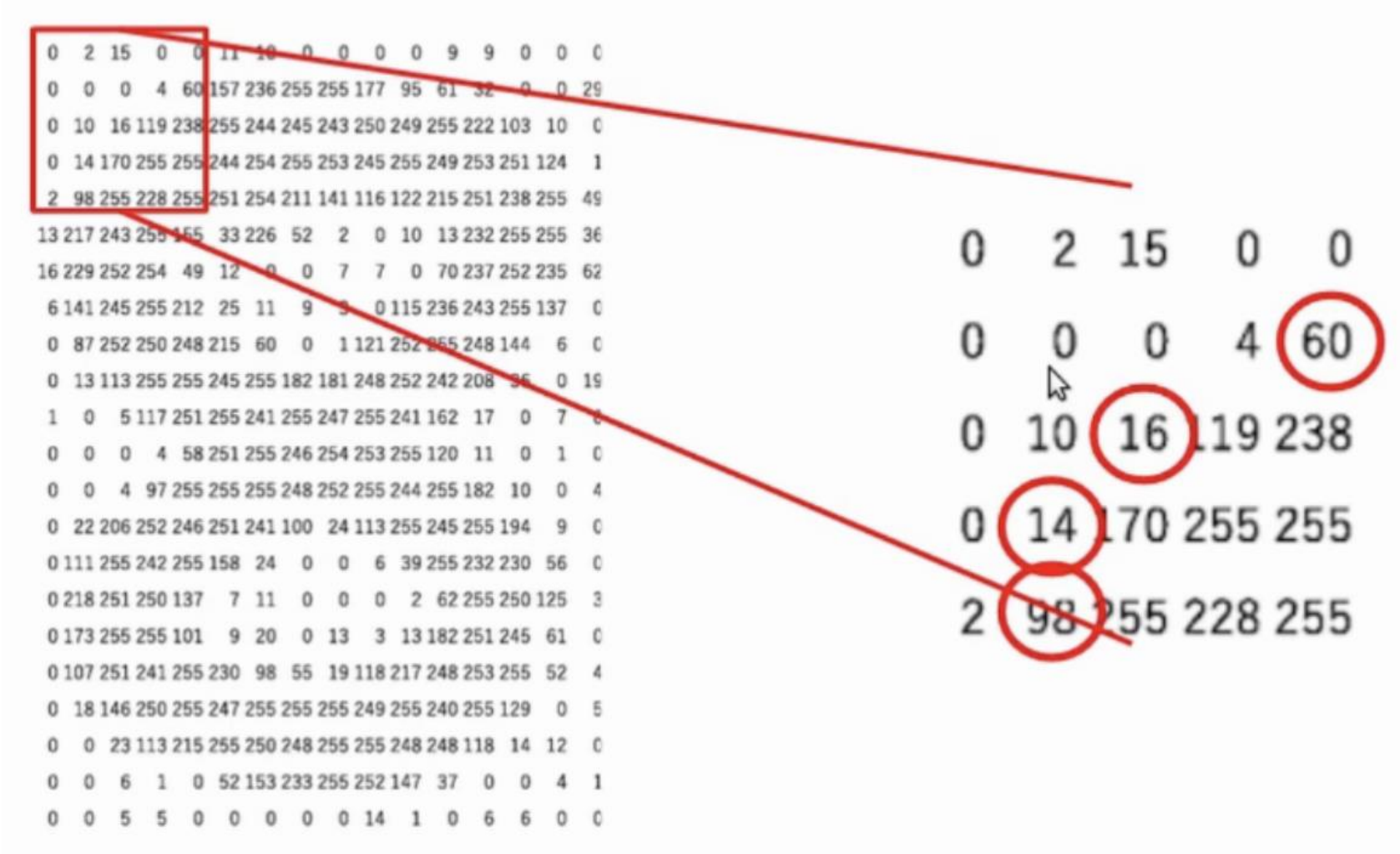


0	2	15	0	0
0	0	0	4	60
0	10	16	119	238
0	14	170	255	255
2	98	255	228	255

-1	0	1
-1	0	1
-1	0	1

(0 x -1) + (0 x -1) + (0 x -1) +
(2 x 0) + (0 x 0) + (10 x 0) +
(15 x 1) + (0 x 1) + (16 x 1)

Edge or No edge



-1	0	1
-1	0	1
-1	0	1

(0 x -1) + (10 x -1) + (14 x -1) +
(0 x 0) + (16 x 0) + (170 x 0) +
(4 x 1) + (119 x 1) + (255 x 1)

Edge Detection process

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

31	111	267	300

-1	0	1
-1	0	1
-1	0	1

(-1x0 + -1x4 + -1x119 + 0x0 + 0x60 +
0x238 + 1x11 + 1x157 + 1x255)

Types of Edge Detection Algorithms

Types

- Laplacian
- Sobel
- Canny

Laplacian

Laplacian edge detection computes gradients

Laplacian(image, CV_64F)

Sobel

Sobel edge detection computes gradient magnitude representations along the x and y axis and then combines both the vertical edges and horizontal edges

Sobel(image, CV_64F, order of derivatives)

Canny

Canny edge detection is a multi-step process

- Blur the image to remove noise
- Compute Sobel gradient images in the x and y direction
- Suppress edges
- Hysteresis thresholding stage that determines if a pixel is "edge-like" or not

Canny(image, non-edge_threshold, edge_threshold)

Contours

What is it?

Contours

A contour is a curve of points, with no gaps in the curve

To find contours in an image:

- Edge detection
- Thresholding

findContours( )

findContours( )

- image
- Types of contour
  - cv2.RETR_EXTERNAL
  - cv2.RETR_LIST
  - cv2.RETR_COMP
  - cv2.RETR_TREE
- approximation of contour
  - cv2.CHAIN_APPROX_SIMPLE
  - cv2.CHAIN_APPROX_NONE
- Returns a tuple
  - output image after applying contour detection
  - cnts list of contours detected
  - hierarchy of the contours

Watershed

What is it?

Watershed

Classic algorithm used for **segmentation** and is especially useful when extracting touching or **overlapping objects** in images

When utilizing the watershed algorithm we must start with **user-defined markers**

These markers can be either defined:

- manually point-and-click
- automatically or heuristically define them
  - thresholding operations
  - morphological operations

Segment & Extract Objects

Import the libraries

Load the image

Apply Pyramid Mean Shift Filtering

Convert the image into grayscale

Threshold the Mean Shifted image

Detect Contours using findContours()

Draw Contours using drawContours()

Display the extracted contours

Shapes

What is it?

Shape detection

Find **contours** of objects

Find **centers** of each object

Recognize **shapes** of each object

- Circle
- Square
- Rectangle
- Triangle
- Pentagon

Label color of a shape

Shape detection

- **Contour Approximation**
  - Algorithm for reducing the number of points in a curve with a reduced set of points
  - Assumption: A curve can be approximated by a series of **short line segments**
- **cv2.approxPolyDP()**
  - contour
  - 1-15% of contour perimeter
- **Shape analysis**
  - If vertices are 3 in the contour
    - Traingle
  - If vertices are 5 in the contour
    - Pentagon
  - If vertices are 4 in the contour
    - If aspect ratio is between 0.95 to 1.05
      - Square
    - If aspect ration is skewed
      - Rectangle

Further Reading

Digital Image Processing 4th edition

Rafael Gonzalez & Richard Woods