Image Analytics

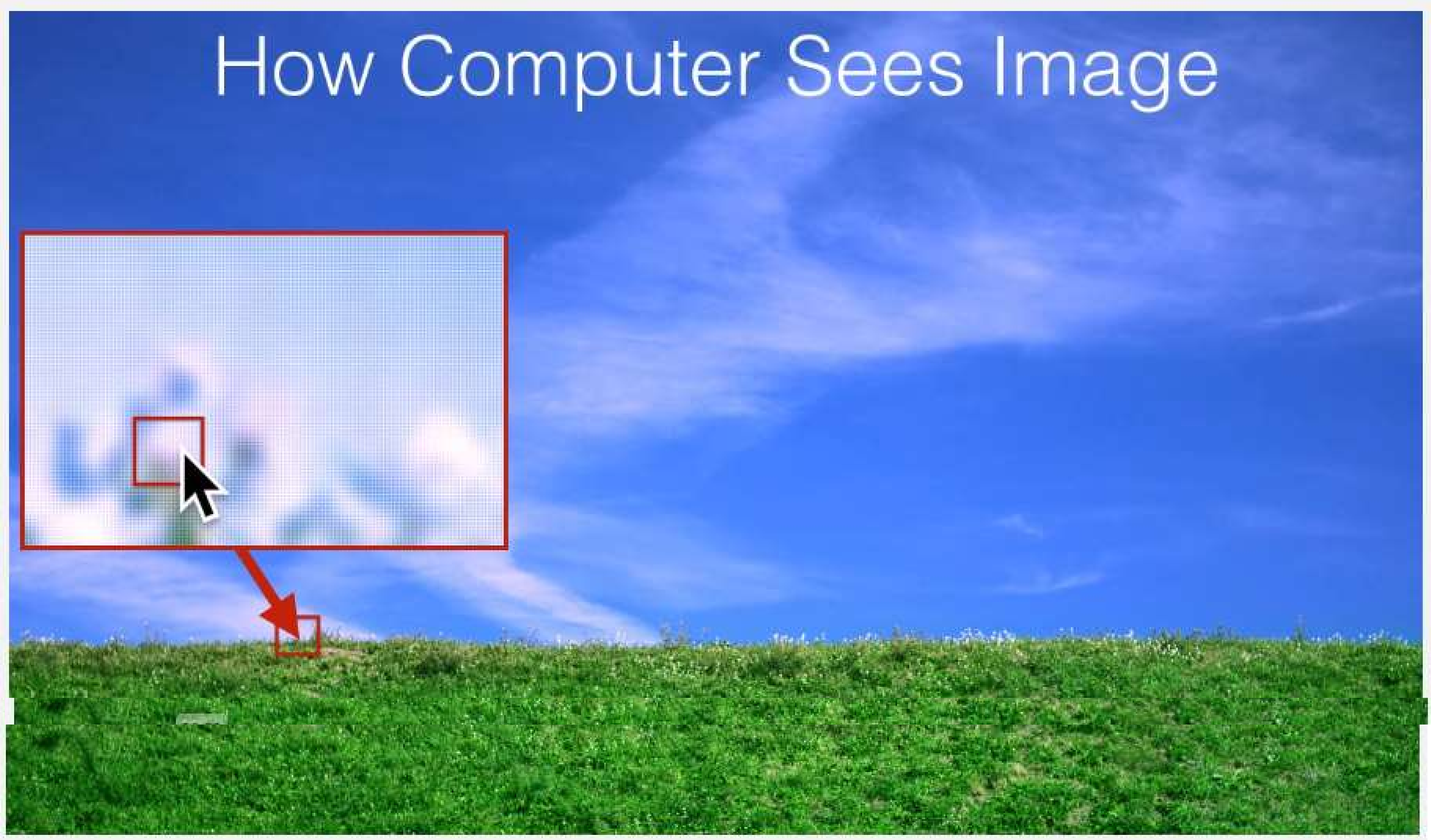
Unit-4

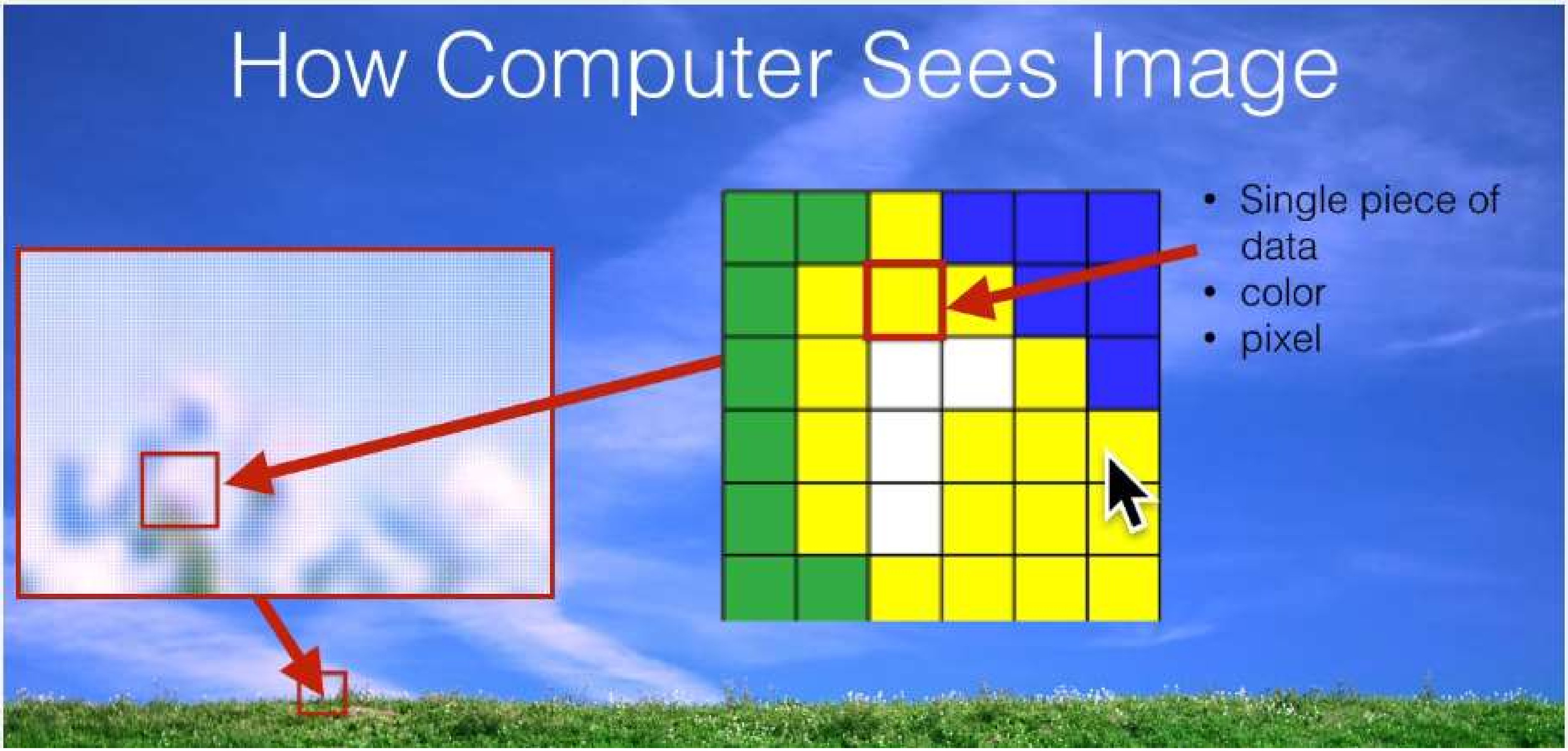
Unit-4 (MCA-A&B)

- **IMAGE ANALYTICS :**Digital Image Representation - Elements of digital image processing, Digital Image Properties-Histograms, Entropy - Relationships between pixels Connectivity, Distance Measures between pixels, Various image formats – bmp, jpeg, tiff, png, gif. Noise in Images – Sources, types, Image Restoration, Image Filtering-Inverse filtering, Wiener Filtering - Segmentation

Lab Exercises:

7 . Digitization and Implementation of Histogram Equalization
8. Implementation of metrics for Noise measures in Image quality
9.Implementation of Image-filtering techniques
10. Implementation of Image classification and Image Clustering


How Computer Sees Image


How Computer Sees Image
• Single piece of data
• color
• pixel

Understanding the pixel grid

- Coordinates
- Grid system

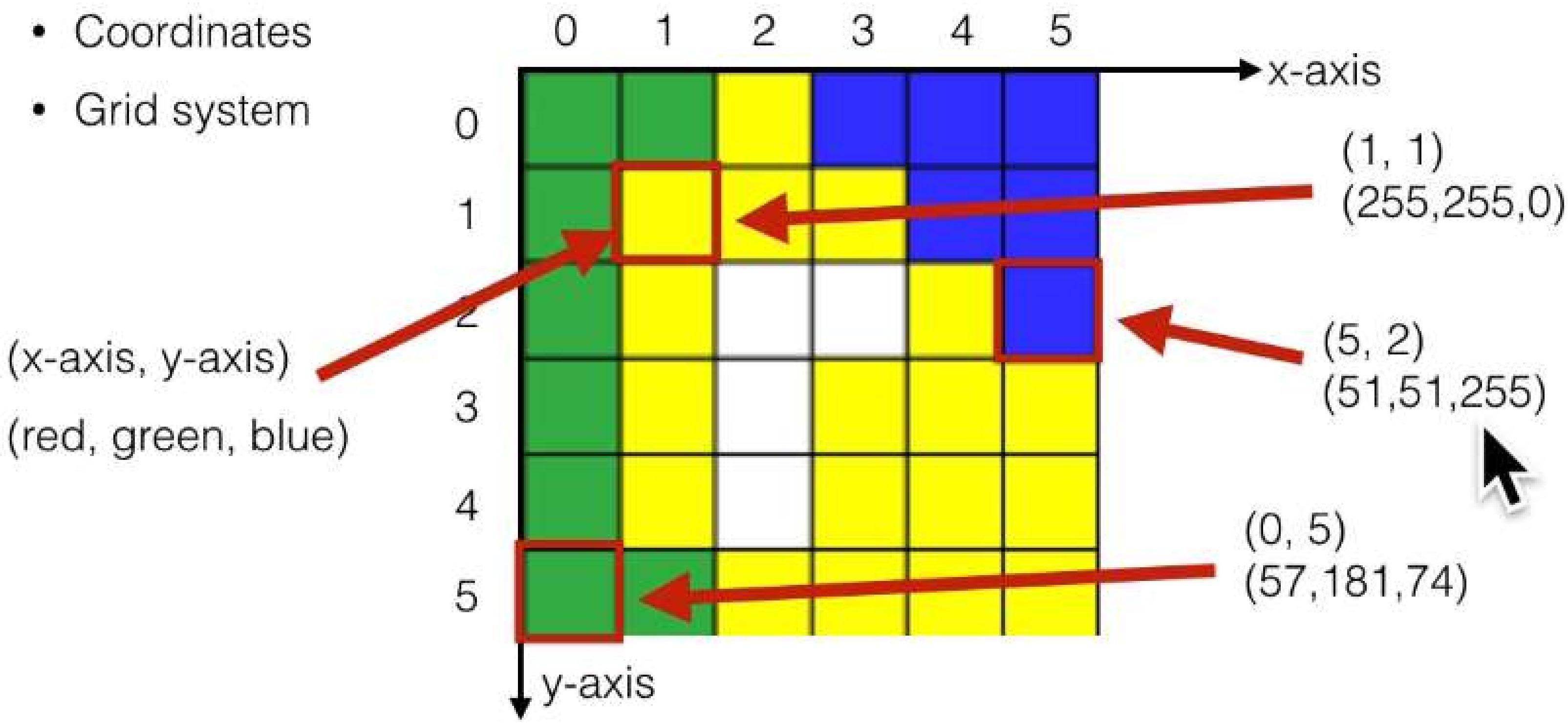


Image Processing

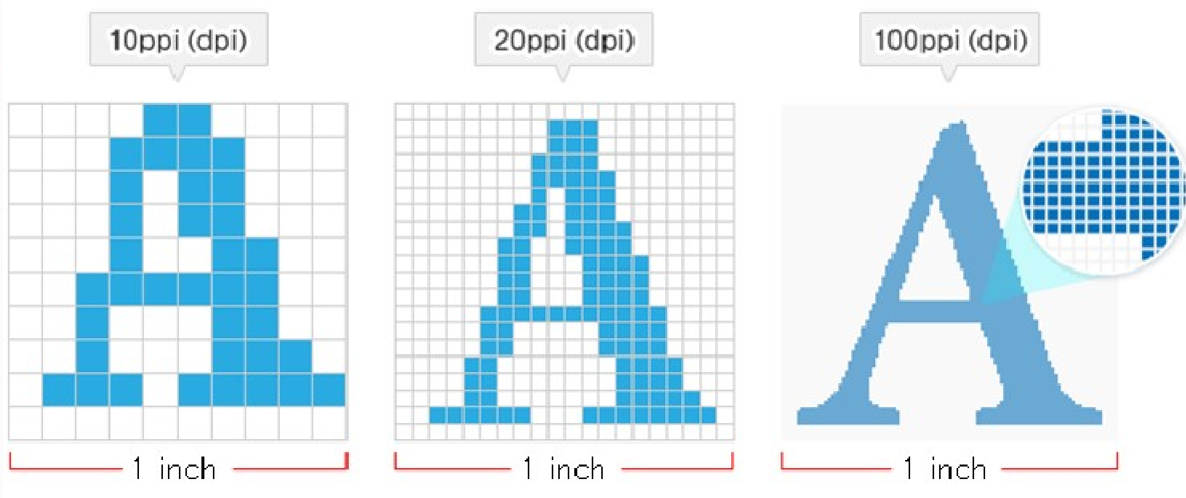
- An image is a visual representation or depiction of an object or scene.
- In the context of digital technology, images are typically stored and processed as a collection of pixels, each representing a small unit of the overall image
- Digital Image processing is the class of methods that deal with manipulating digital images through the use of computer algorithms. It is an essential preprocessing step in many applications, such as face recognition, object detection, and image compression

Key Concepts in Image Representation

- **Pixels**:The basic unit of an image is the pixel (picture element), which represents the smallest component of the image. Each pixel has a value that describes its intensity or color.
- **Resolution:**Resolution refers to the number of pixels in an image. It is usually described in terms of width × height (e.g., 1920 × 1080 pixels).
- **Color Depth**:Color depth, or bit depth, indicates the number of bits used to represent the color of each pixel. Common color depths include 8-bit (256 colors), 16-bit, 24-bit (true color), and 32-bit (with alpha channel for transparency).

Key Properties and Concepts of an Image

- **Pixel:** A pixel (short for picture element) is the smallest unit of an image. It is a single point in a raster image and contains information about color and brightness. The combination of pixels creates the overall image.
- **Resolution:** The resolution of an image refers to the number of pixels it contains. Higher resolution images have more pixels, resulting in greater detail and clarity. Common units for resolution include pixels per inch (PPI) or dots per inch (DPI).

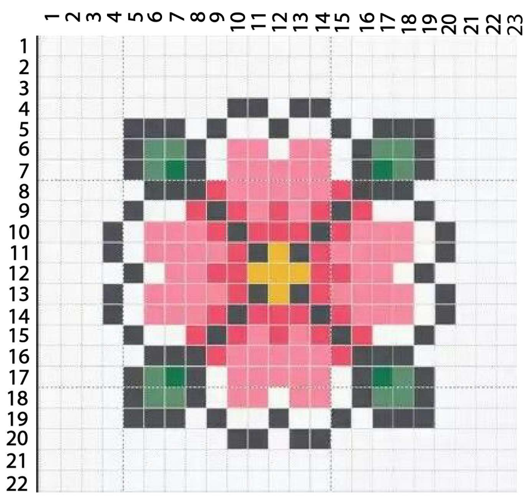


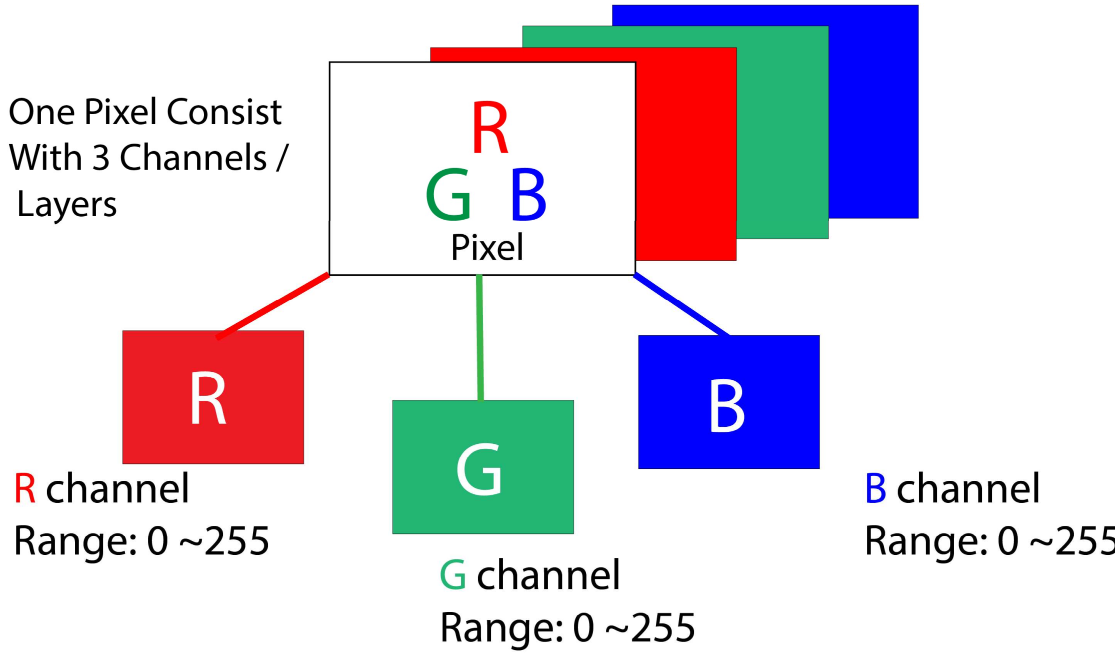
Key Properties and Concepts of an Image Cont..

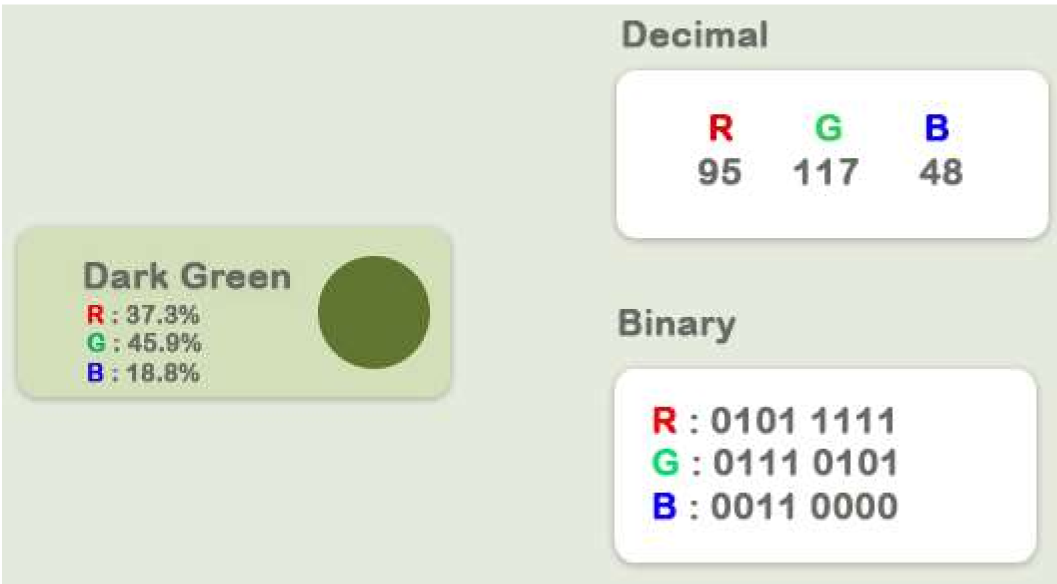
- **Color Depth:** Color depth, also known as bit depth, represents the number of bits used to represent the color of each pixel in an image. Common color depths include 8-bit (256 colors), 16-bit (65,536 colors), and 24-bit (16.7 million colors).

Each pixel coordinate (x, y) contains 3 values ranging for intensities of 0 to 255 (8-bit)
- Red - Green - Blue

Mixing different intensities of each color gives us the full color spectrum.

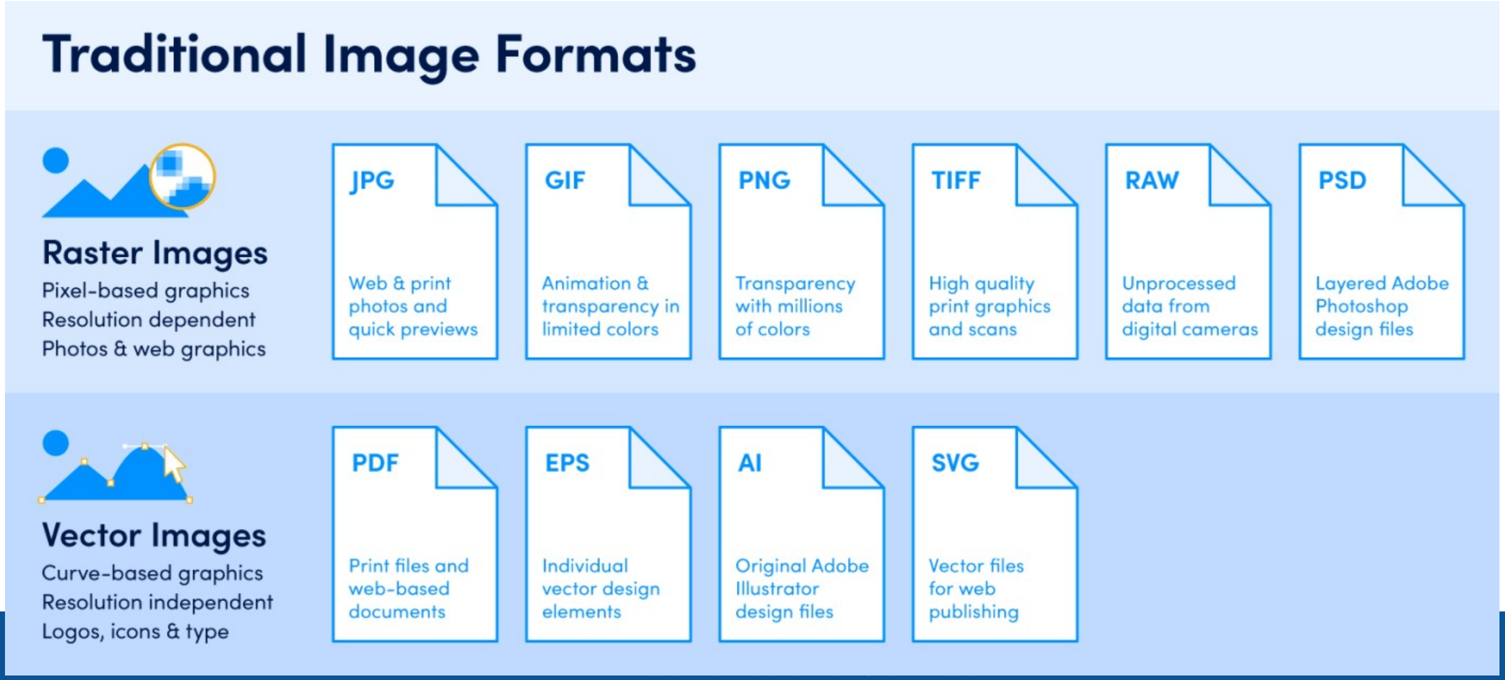





Dark Green
R : 37.3%
G : 45.9%
B : 18.8%
Decimal
R G B
95 117 48
Binary
R : 0101 1111
G : 0111 0101
B : 0011 0000

Key Properties and Concepts of an Image Cont..

- **File Format:** Images are stored in various file formats, each with its own characteristics. Common formats include JPEG, PNG, GIF, BMP, and TIFF. Different formats have different compression methods, support for transparency, and other features.
- **Aspect Ratio:** The aspect ratio is the ratio of the width to the height of an image. Common aspect ratios include 4:3 (standard television), 16:9 (widescreen television), and 3:2 (common in photography).



Types of Image Representation

Binary Images:

- Binary images are a type of grayscale image where each pixel can be one of two values: 0 (black) or 1 (white). They are often used in document scanning and certain types of image analysis

Grayscale Images:

- Grayscale images are composed of shades of gray, varying from black (0) to white (255) in an 8-bit image. Each pixel has a single value representing the intensity of light.

Color Images:

- Color images contain multiple channels representing different color components. The most common color model is the RGB (Red, Green, Blue) model, where each pixel is represented by three values corresponding to the intensities of red, green, and blue

Types of Image Representation

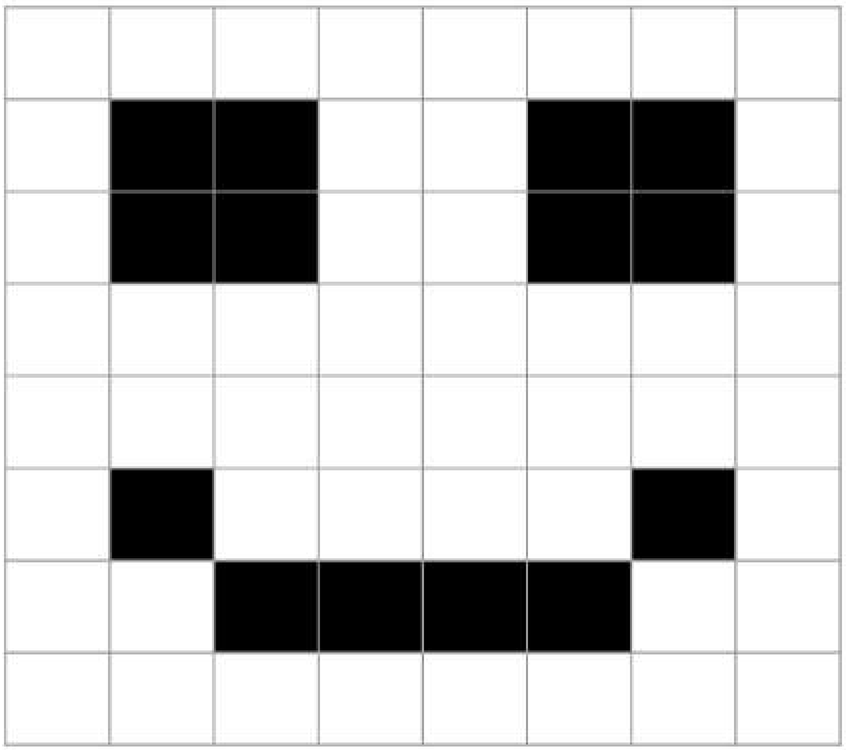
Indexed Images:

- Indexed images use a colormap or palette to map pixel values to colors. This representation is efficient in terms of storage, especially for images with a limited number of colors.

Vector Images:

- Unlike raster images (pixel-based), vector images use geometric shapes like points, lines, and curves to represent images. They are resolution-independent and are commonly used in graphic design and typography

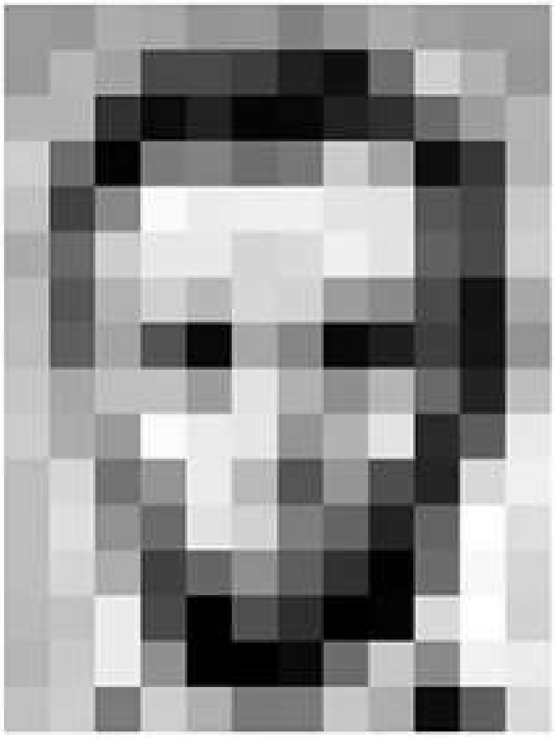
Binary Images

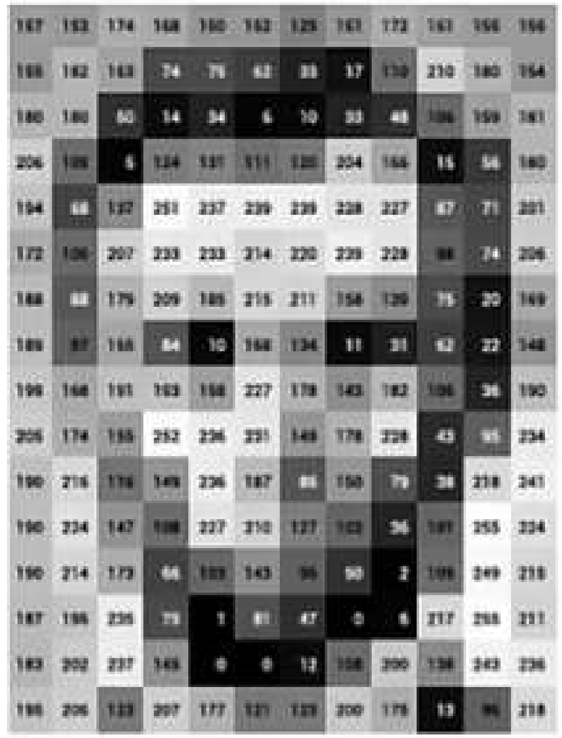


0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0

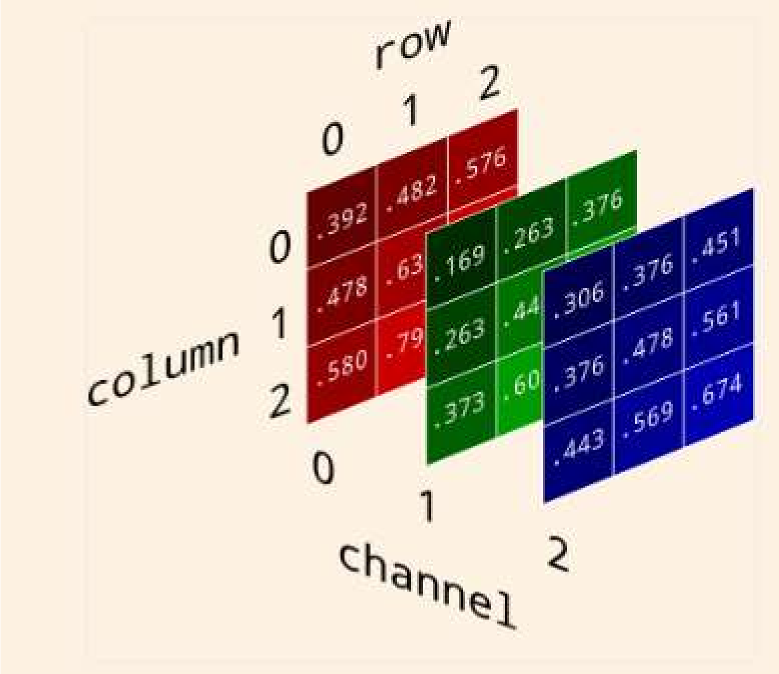
The binary for the image above would be 00000000 01100110 01100110 00000000 00000000 01000010 00111100 00000000

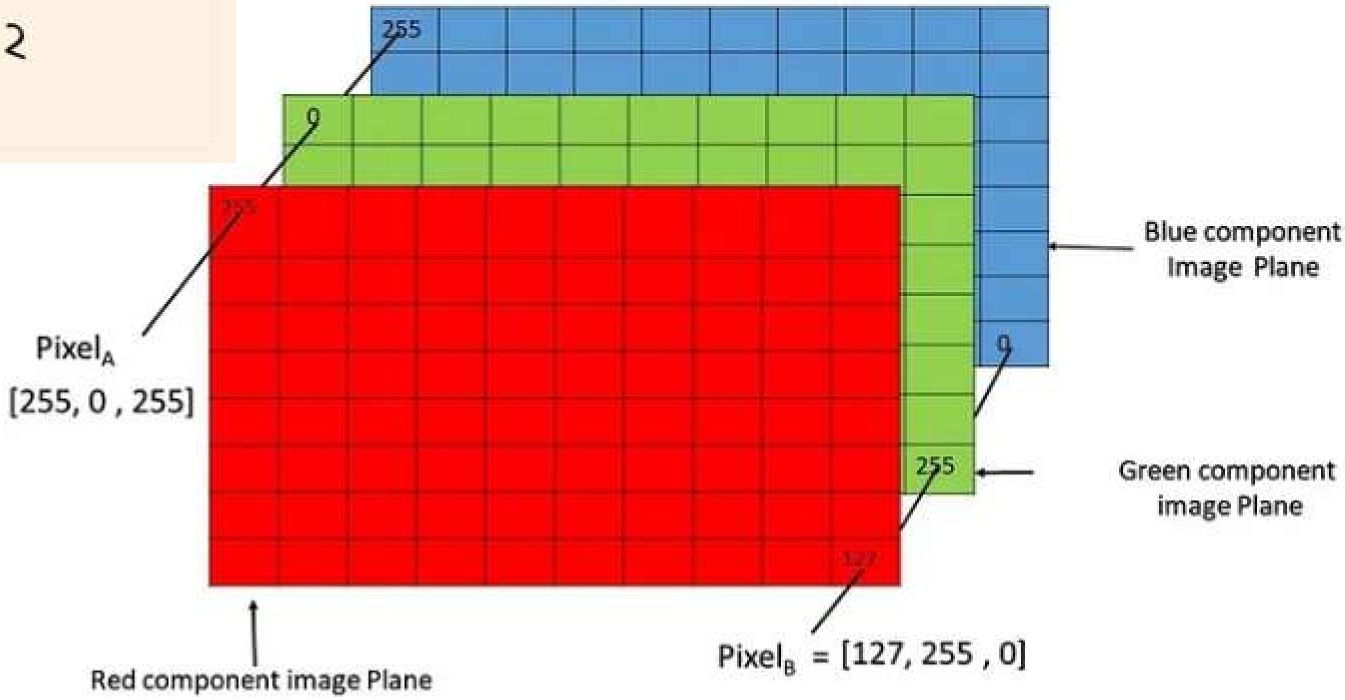
Gray Scale Image



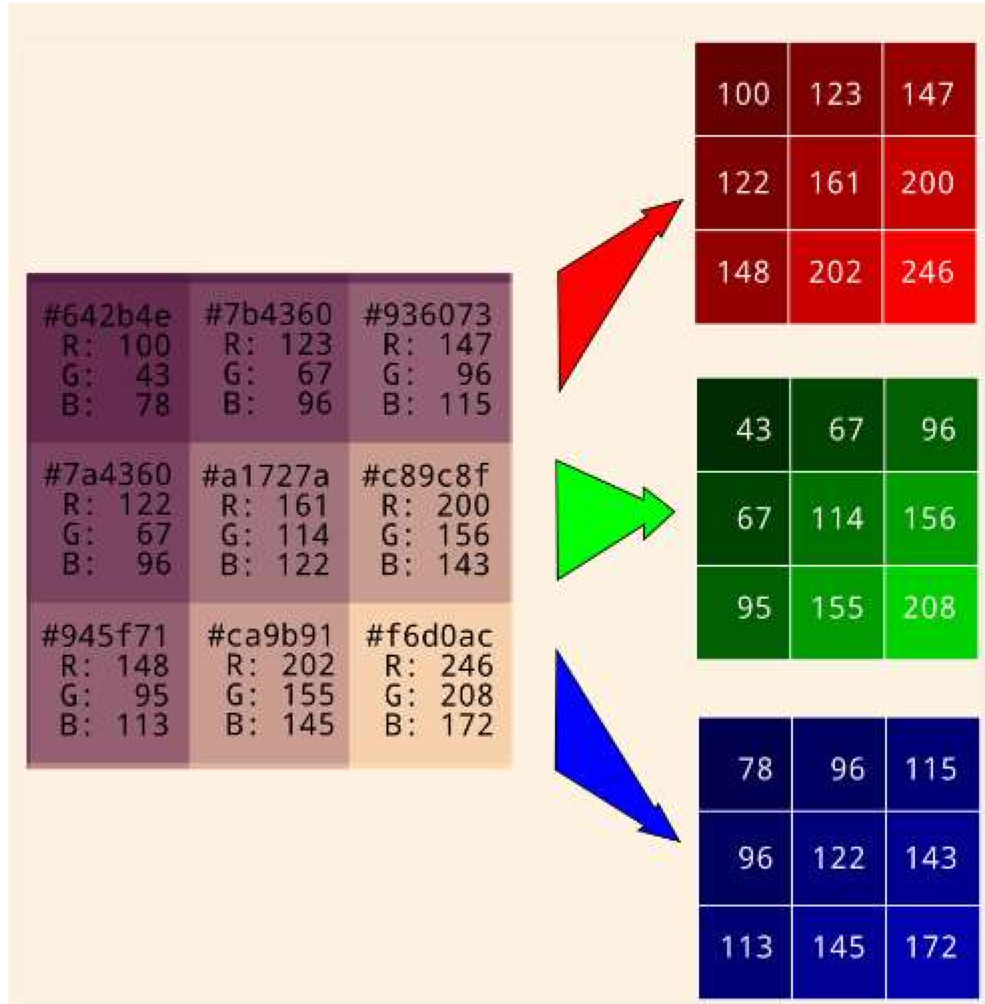


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218


row
0 1 2
column
0 1 2
.392 .482 .576
.478 .63
.580 .79
.169 .263 .376
.263 .44
.373 .60
.306 .376 .451
.376 .478 .561
.443 .569 .674
0 1 2
channel

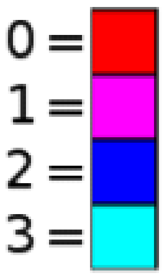

255
0
0
255
Blue component
Image Plane
Green component
image Plane
$Pixel_A$
[255, 0 , 255]
Red component image Plane
$Pixel_B$ = [127, 255 , 0]

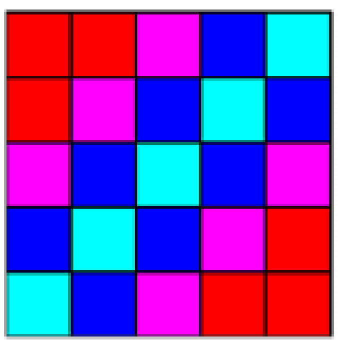
red	#ff0000	
green	#00ff00	
blue	#0000ff	
yellow	#ffff00	
cyan	#00ffff	
magenta	#ff00ff	
black	#000000	
gray	#777777	
white	#ffffff	

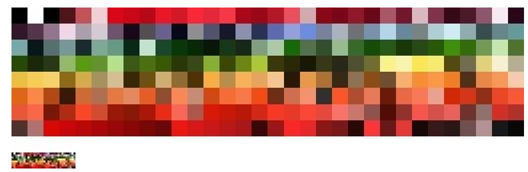


Indexed Images

0	0	1	2	3
0	1	2	3	2
1	2	3	2	1
2	3	2	1	0
3	2	1	0	0







Vector Images

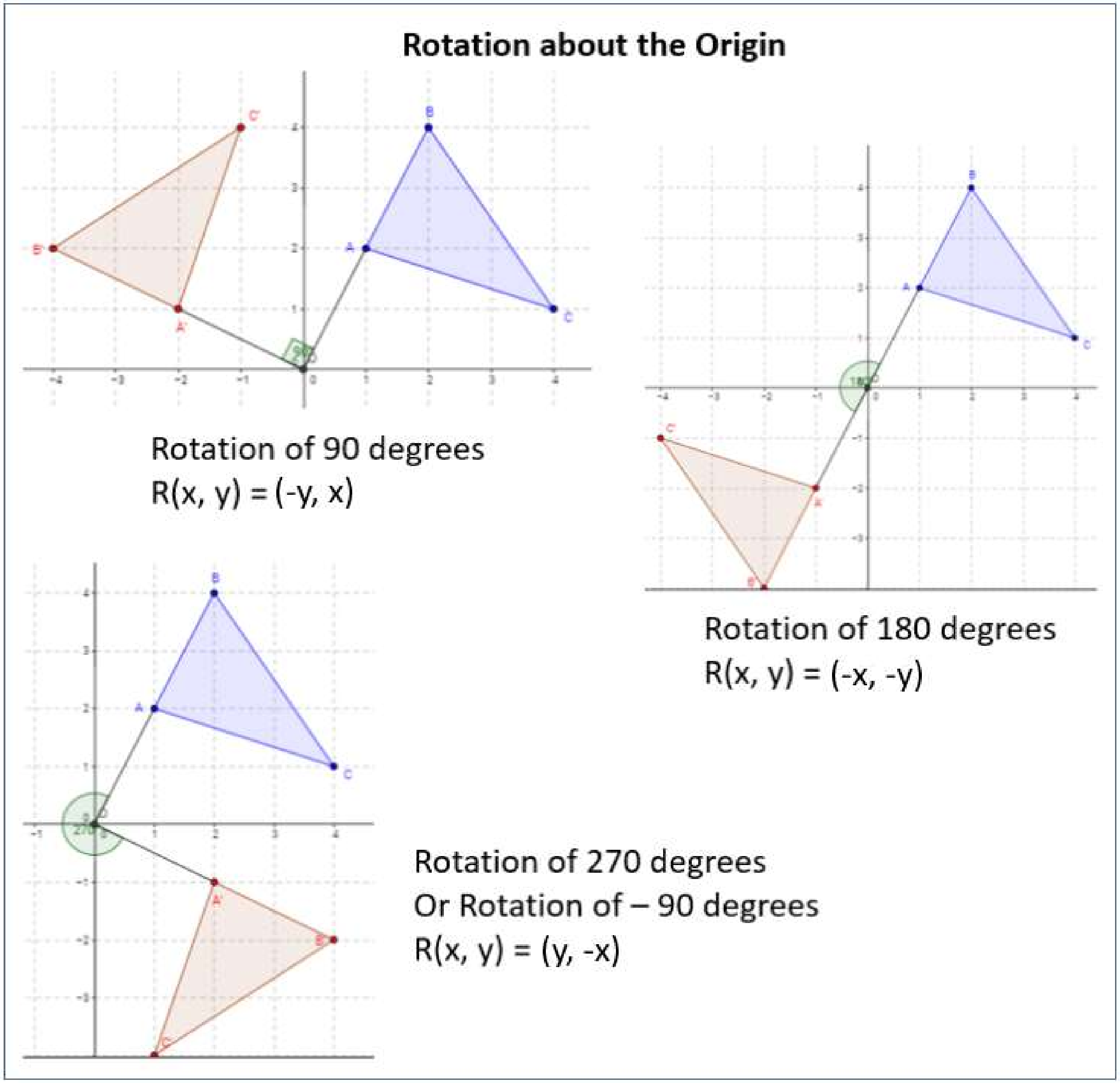
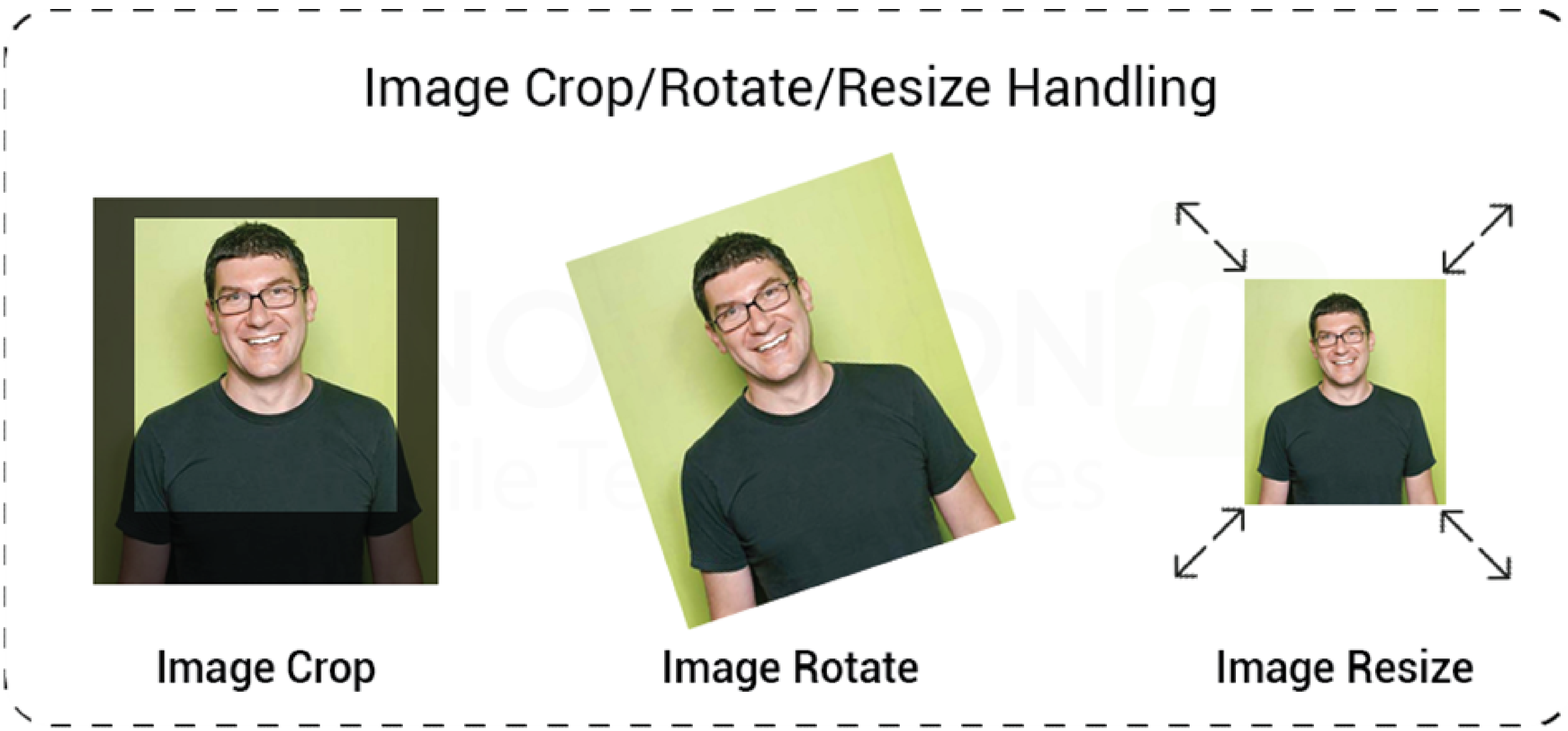


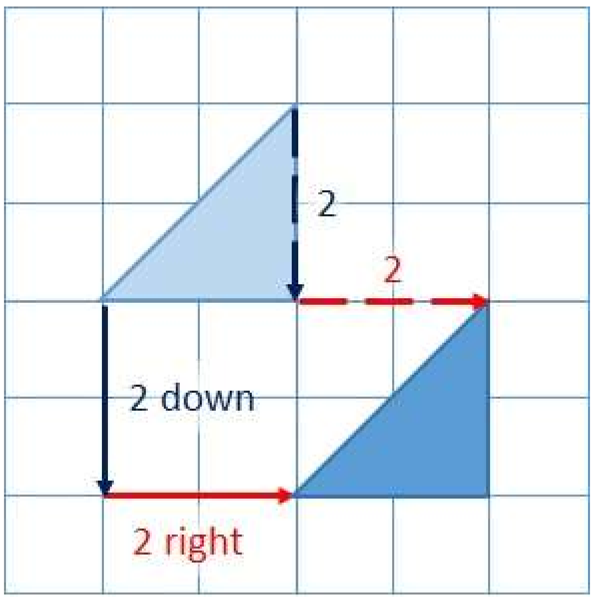
Image processing tasks

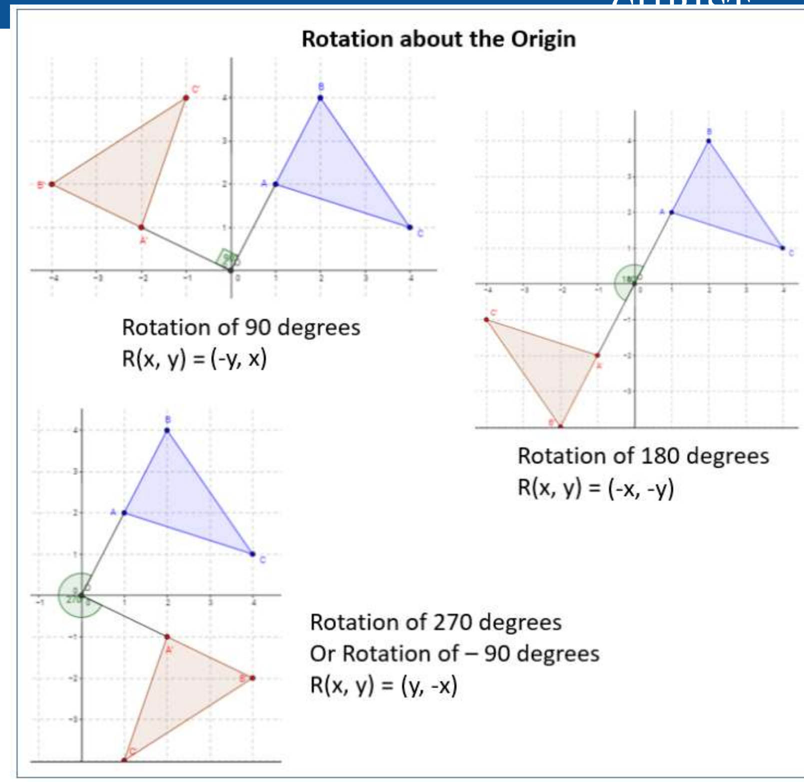
- Reading and writing images
- Image manipulation (resizing, cropping, rotating, etc.)
- Color space conversion
- Geometric Transformations (Scaling, Translation, Rotation, Transformation,
- Filtering
- Edge detection
- Image Enhancement(Contrast enhancement, Noise reduction, Sharpening, Color manipulation)

Image manipulation

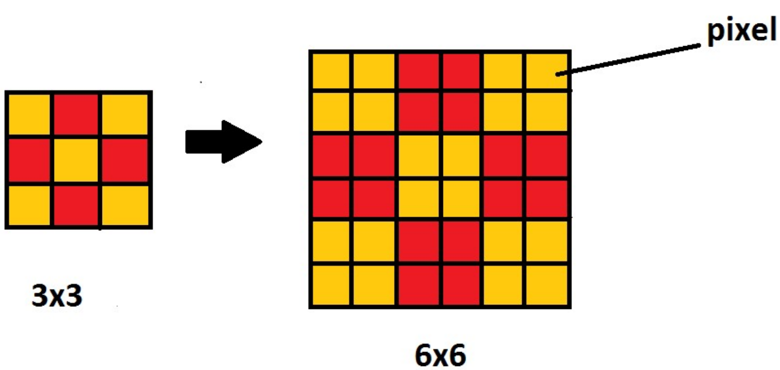


Translation

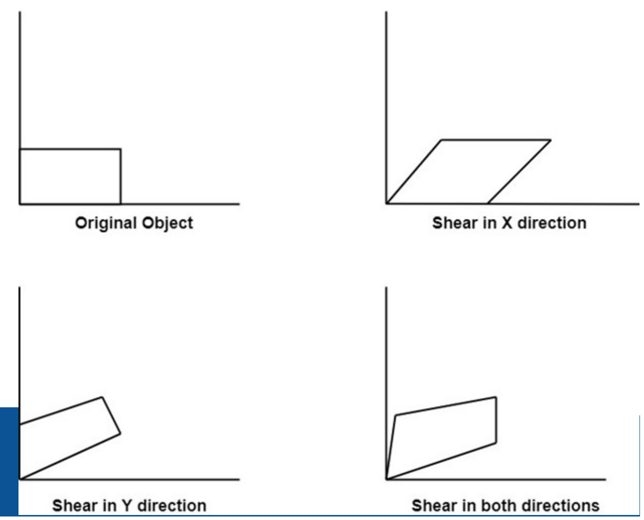


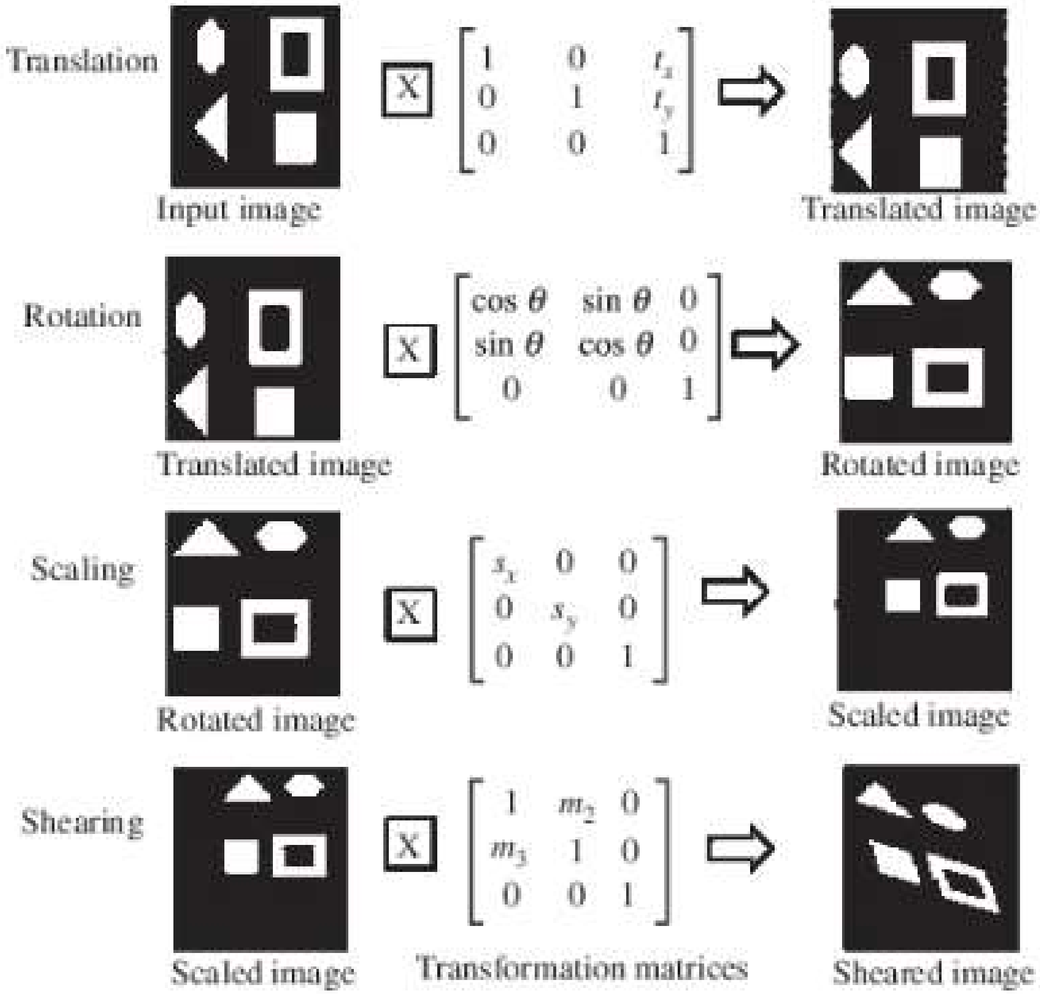


Scaling



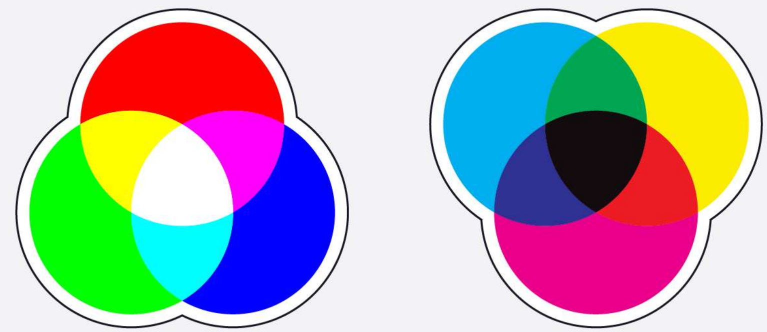
Shearing





$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & m_2 & 0 \\ m_3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Color space conversion



RGB Mode

- Colors: red, green, blue
- Additive mixing
- Mix to create white
- 16.7 million color possibilities
- Smaller file sizes

CMYK Mode

- Colors: cyan, magenta, yellow
- Subtractive mixing
- Mix to create black
- 16 thousand color possibilities
- Larger file sizes

Image Preprocessing

- Resizing: Adjust the image size to meet the input requirements of the learning model.
- Normalization: Scale pixel values to a standard range (e.g., [0, 1] or [-1, 1]) to enhance model convergence.
- Normalization: Standardize pixel values by subtracting mean and dividing by standard deviation.
- Data Augmentation: Generate new training samples by applying random transformations like rotation, flipping, and zooming to increase dataset diversity.
- Cropping: Crop images to focus on the region of interest or to obtain consistent input sizes.

Image Preprocessing(Cont..)

- Gray Scaling: Convert color images to grayscale, reducing computational complexity and focusing on intensity information.
- Color Space Conversion: Convert images between color spaces (e.g., RGB to HSV) to emphasize or extract specific color information.
- Contrast Adjustment: Adjust image contrast to enhance or normalize brightness levels.
- Histogram Equalization: Enhance image contrast by equalizing the pixel intensity distribution.
- Gaussian Blurring: Apply a Gaussian filter to smooth images, reducing noise and preserving important details.
- Noise Reduction: Remove or reduce noise in images through techniques like median filtering or denoising algorithms.
- Edge Detection: Highlight edges in images using techniques like the Sobel or Canny edge detectors.

Image Preprocessing(Cont..)

- Thresholding: Convert images to binary format by setting a threshold, useful for segmenting objects from the background.
- Morphological Transformations: Perform operations like dilation and erosion to manipulate image structures.
- Image Inversion: Invert pixel values to highlight different aspects of the image.
- Centering: Center images or objects within the frame for improved consistency.
- Hue, Saturation, and Value (HSV) Adjustment: Modify color components in the HSV color space to control brightness, saturation, and hue.
- https://medium.com/@nimritakoul01/image-processing-using-opencv-python-9c9b83f4b1ca#:~:text=In%20OpenCV%2C%20images%20are%20represented,of%20color%20for%20the%20image.

Why Image Preprocessing

- Image enhancement: To make images more readable for machines or humans. E.g., improving brightness, contrast, color balancing or correction.
- Image restoration: To recover the obscure parts of image like those caused by motion blur, noise etc.
- Segmentation: Partitioning an image into multiple objects present in the image.
- Representation and description of objects in an image: based on boundaries or pixel values.
- Object detection and recognition

Image Formats - – bmp, jpeg, tiff, png, gif

- Image formats are essential for storing and managing images in digital form. Each format has unique characteristics and uses, making them suitable for different applications in image processing. Common image formats: BMP, JPEG, TIFF, PNG, and GIF.
- Each image format serves different purposes in image processing:
    - **BMP**: High-quality, uncompressed images, large file size.
    - **JPEG**: Compressed images, adjustable quality, widely used for photos.
    - **TIFF**: Flexible, high-quality, suitable for professional use.
    - **PNG**: Lossless compression, supports transparency, ideal for web graphics.
    - **GIF**: Lossless, supports animations, limited color range, used for simple graphics and web animations.

BMP (Bitmap)

- **Pixel Storage**: BMP files store pixel data in a straightforward, uncompressed format.
- **Representation**: Each pixel's color is typically represented in a fixed number of bits. For example, a 24-bit BMP file uses 8 bits for each of the red, green, and blue color channels, resulting in a total of 24 bits per pixel.
- **Structure**: Pixels are usually stored in a row-major order, from left to right and top to bottom

JPEG (Joint Photographic Experts Group)

- **Pixel Storage**: JPEG uses lossy compression, which transforms pixel data into frequency domain components using the Discrete Cosine Transform (DCT).
- **Representation**: Pixel colors are not stored directly. Instead, the image is divided into 8x8 blocks, and the color information is represented in terms of DCT coefficients, which are then quantized and encoded.
- **Structure**: The original pixel representation is reconstructed during decompression, but some data is lost, leading to compression artifacts.

TIFF (Tagged Image File Format)

- **Pixel Storage**: TIFF supports various compression methods (lossy and lossless) and different pixel representations.
- **Representation**: Pixels can be stored in different formats, such as uncompressed, LZW compressed, or JPEG compressed. TIFF can handle multiple color depths and color spaces.
- **Structure**: Pixels can be stored in strips or tiles, and the format supports multiple images (pages) in a single file.

PNG (Portable Network Graphics)

- **Pixel Storage**: PNG uses lossless compression, specifically the Deflate algorithm.
- **Representation**: Each pixel is represented without any loss of information. PNG supports various bit depths (e.g., 8, 16 bits per channel) and an alpha channel for transparency.
- **Structure**: Pixels are stored in a row-major order with optional interlacing (Adam7).

GIF (Graphics Interchange Format)

- **Pixel Storage**: GIF uses lossless compression with the LZW algorithm.
- **Representation**: Pixels are represented using a limited color palette (up to 256 colors) chosen from the full 24-bit RGB color space.
- **Structure**: Pixels are stored using a color index table (palette), which references

Pixel Representation Differences

- **Color Depth**:
    - **BMP**: Fixed color depth (e.g., 24-bit, 8-bit).
    - **JPEG**: Variable color depth, typically stored as 24-bit but compressed.
    - **TIFF**: Variable color depth, supports multiple depths.
    - **PNG**: Variable color depth, supports high bit depths (e.g., 24-bit, 48-bit).
    - **GIF**: Limited to 256 colors, using a palette.
- **Compression**:
    - **BMP**: Typically uncompressed.
    - **JPEG**: Lossy compression.
    - **TIFF**: Both lossy and lossless options.
    - **PNG**: Lossless compression.
    - **GIF**: Lossless compression with color limitations.

Pixel Representation Differences

- **Transparency**:
  - **BMP**: Limited or no support.
  - **JPEG**: No support.
  - **TIFF**: Supports transparency via alpha channels.
  - **PNG**: Supports transparency via alpha channels.
  - **GIF**: Supports simple binary transparency (one color can be transparent).
- **Animation**:
  - **BMP**: No support.
  - **JPEG**: No support.
  - **TIFF**: Limited support (multi-page TIFFs).
  - **PNG**: No support (APNG is an extension for animation).
  - **GIF**: Supports animation.

Run-length encoding

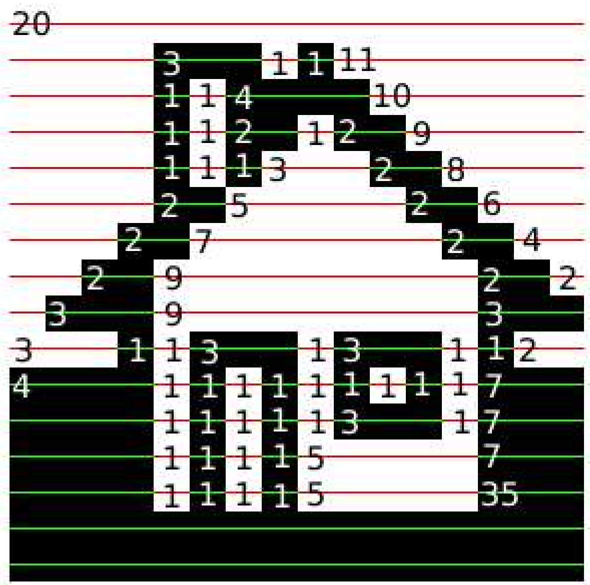


Image Processing Python Libraries

List of some popular image processing Python libraries:

- OpenCV
- Scikit-image
- SciPy
- Pillow (PIL Fork)
- Mahotas
- SimpleCV
- Matplotlib
- Numpy

Digital Image Properties: Histograms and Entropy

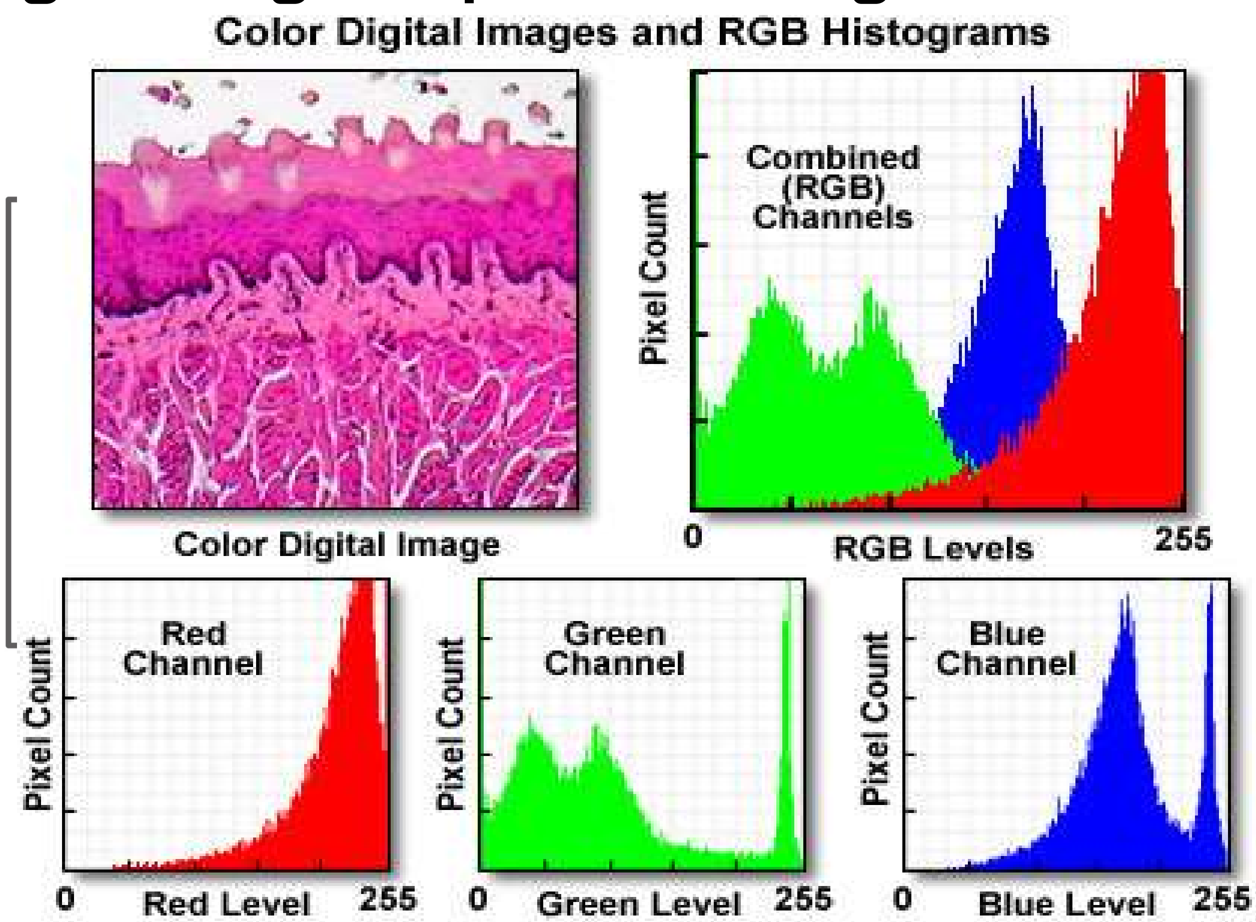
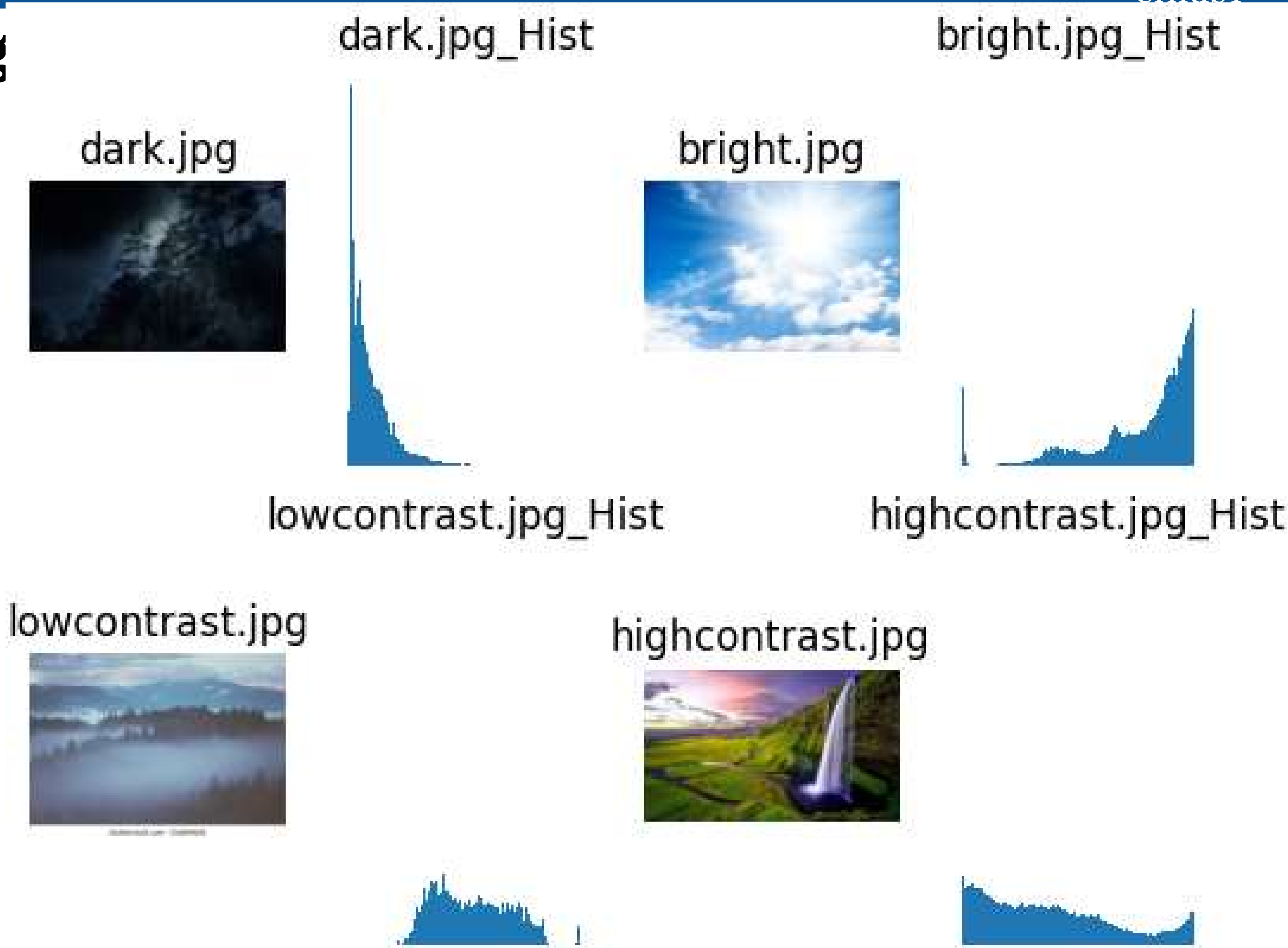


Figure 8

- **Interpretation** Histograms help in understanding the contrast and brightness of an image. A well-spread histogram indicates good contrast, while a histogram concentrated in a narrow range indicates low contrast.

Histog


dark.jpg_Hist
bright.jpg_Hist
dark.jpg
bright.jpg
lowcontrast.jpg_Hist
highcontrast.jpg_Hist
lowcontrast.jpg
highcontrast.jpg

Entropy of Image

- Probability of each pixel is calculated as follows

Symbol	1	2	3	4	5	6	7	8	9
Probability	$\frac{2}{16}$ ✓	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{2}{16}$	$\frac{2}{16}$	$\frac{2}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{2}{16}$

1	2	3	5
3	4	4	6
9	9	8	7
1	3	5	6

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$$Entropy = -\{\frac{2}{16}\log_2\frac{2}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{3}{16}\log_2\frac{3}{16} + \frac{2}{16}\log_2\frac{2}{16} + \frac{2}{16}\log_2\frac{2}{16} + \frac{2}{16}\log_2\frac{2}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{2}{16}\log_2\frac{2}{16}\}$$

- **Interpretation:** Entropy gives an idea about the complexity of the image. Higher entropy values indicate more detailed and complex images, while lower entropy values indicate simpler, more uniform images.

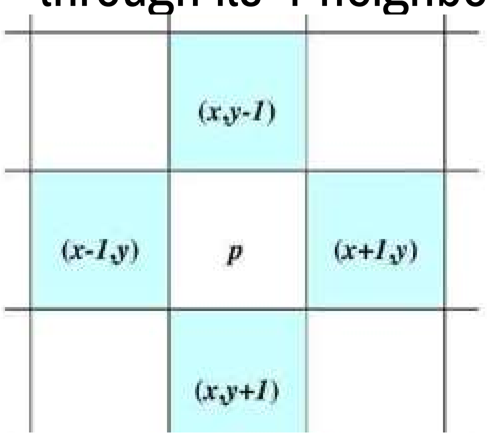
Relationships between pixels Connectivity

1. Neighbors of a Pixel:

- The neighboring pixels of a given pixel can be classified based on their relative positions.

Types of Neighborhoods:

- **4-Connected Neighborhood:**Includes the four horizontal and vertical neighbors of a pixel.
- **8-Connected Neighborhood:**Includes the four diagonal neighbors in addition to the 4-neighbors.
- **D-Connected Neighborhood (Diagonal-Connected):**Includes the 4-neighbors and only those diagonal neighbors that are connected to the pixel through its 4-neighbors.



4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{array}{c} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{array} \right\}$$

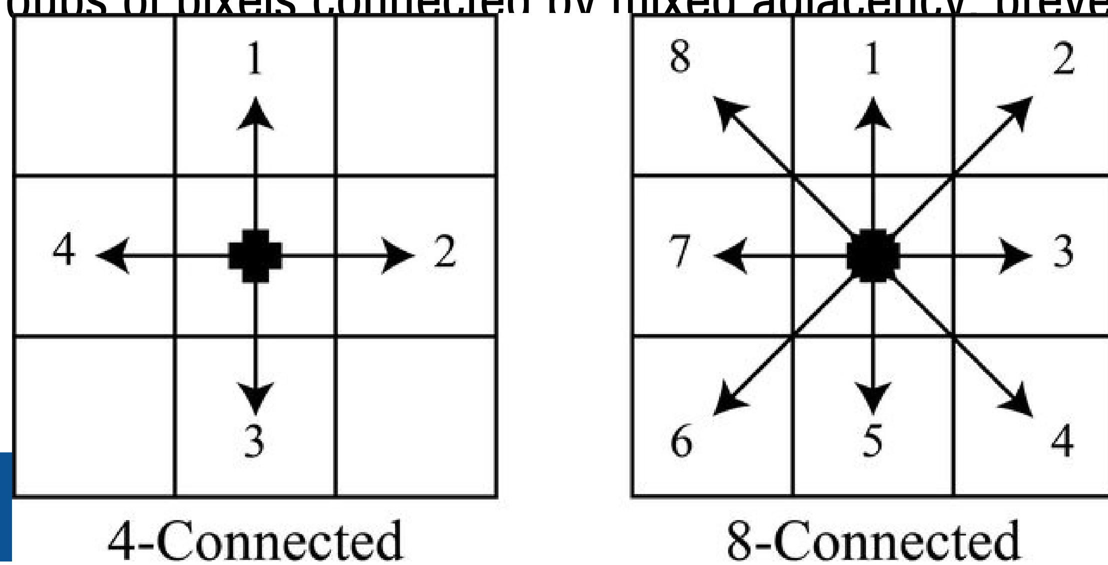
Relationships between pixels Connectivity

2. Adjacency:

- **4-Adjacency**: Two pixels are 4-adjacent if one is a 4-neighbor of the other.
- **8-Adjacency**: Two pixels are 8-adjacent if one is an 8-neighbor of the other.
- **m-Adjacency (Mixed Adjacency)**: Combines 4-adjacency and diagonal adjacency. Used to avoid ambiguities in connected component analysis

3. Connectivity:

- Connectivity defines how pixels are grouped together based on adjacency.
- **4-Connectivity**: Groups of pixels connected by 4-adjacency.
- **8-Connectivity**: Groups of pixels connected by 8-adjacency.
- **m-Connectivity**: Groups of pixels connected by mixed adjacency, preventing ambiguities.



Relationships between pixels Connectivity

4. Path:

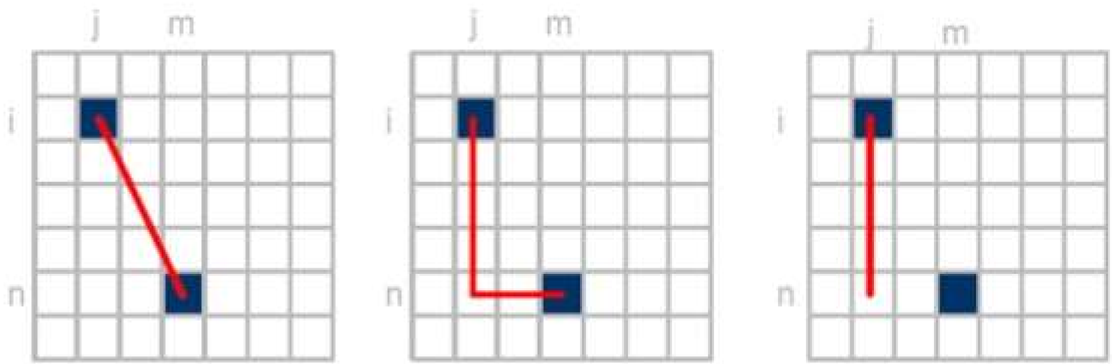
- A path between two pixels p and q is a sequence of pixels where each consecutive pair is adjacent.
- **4-Path**: Sequence of pixels connected by 4-adjacency.
- **8-Path**: Sequence of pixels connected by 8-adjacency.
- **m-Path**: Sequence of pixels connected by mixed adjacency.

5. Regions and Boundaries:

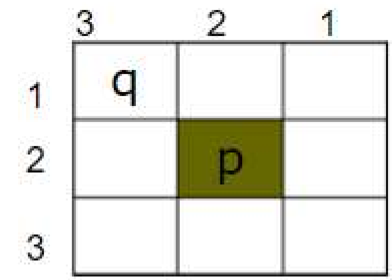
- **Region**: A set of connected pixels with the same properties (e.g., intensity values).
- **Boundary (or Border)**: The set of pixels that separate a region from its background or other regions.

https://cuitutorial.com/basic-relationships-between-pixels/

Distance Measure



Euclidean Distance	City Block Distance	Chessboard Distance
$=\sqrt{(i-n)^2 + (j-m)^2}$	$= \lvert i-n\rvert + \lvert j-m\rvert$	$= \max[\, \lvert i-n\rvert, \lvert j-m\rvert \,]$



Euclidian distance : ((1-2)^2(1-2)+^{2}($^{1/2}$.sqrt(2) =

D4(City Block distance): |1-2| +|1-2| =2

D8(chessboard distance) : max(|1-2|,|1-2|)= 1

- Relationships between pixels Connectivity, Distance Measures between pixels
- https://cuitutorial.com/basic-relationships-between-pixels/

Histogram Equalization

- Class Room teaching

Applications

- **Image blending**

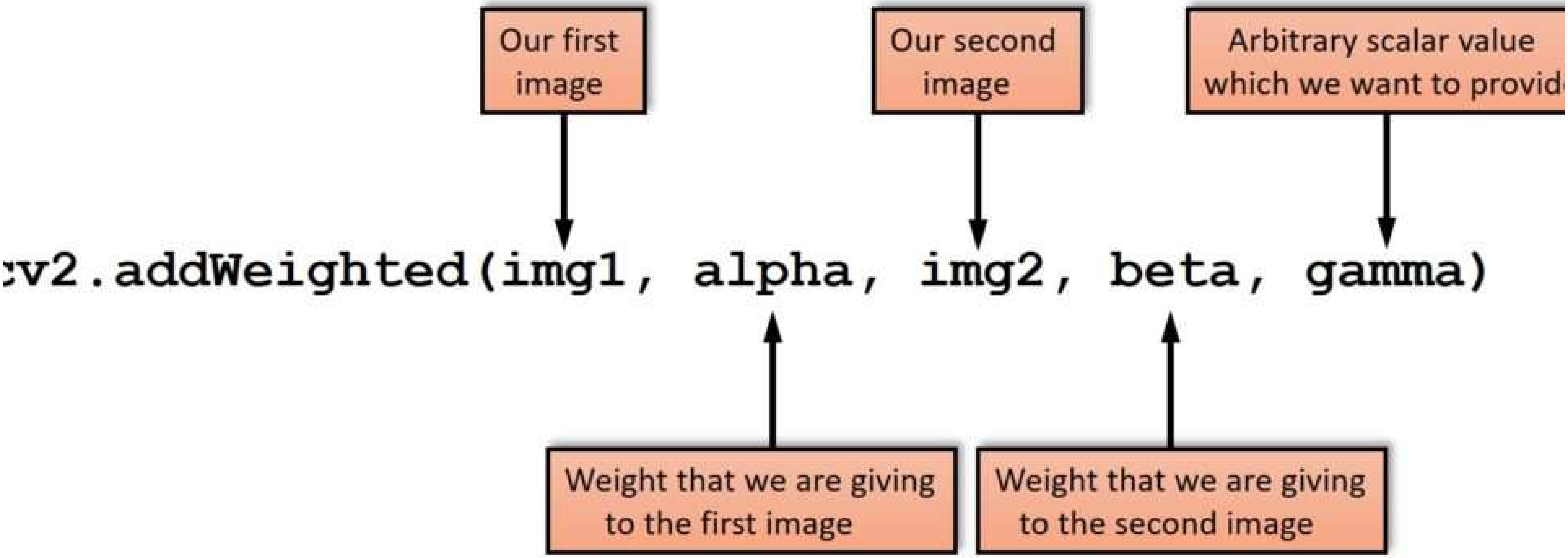
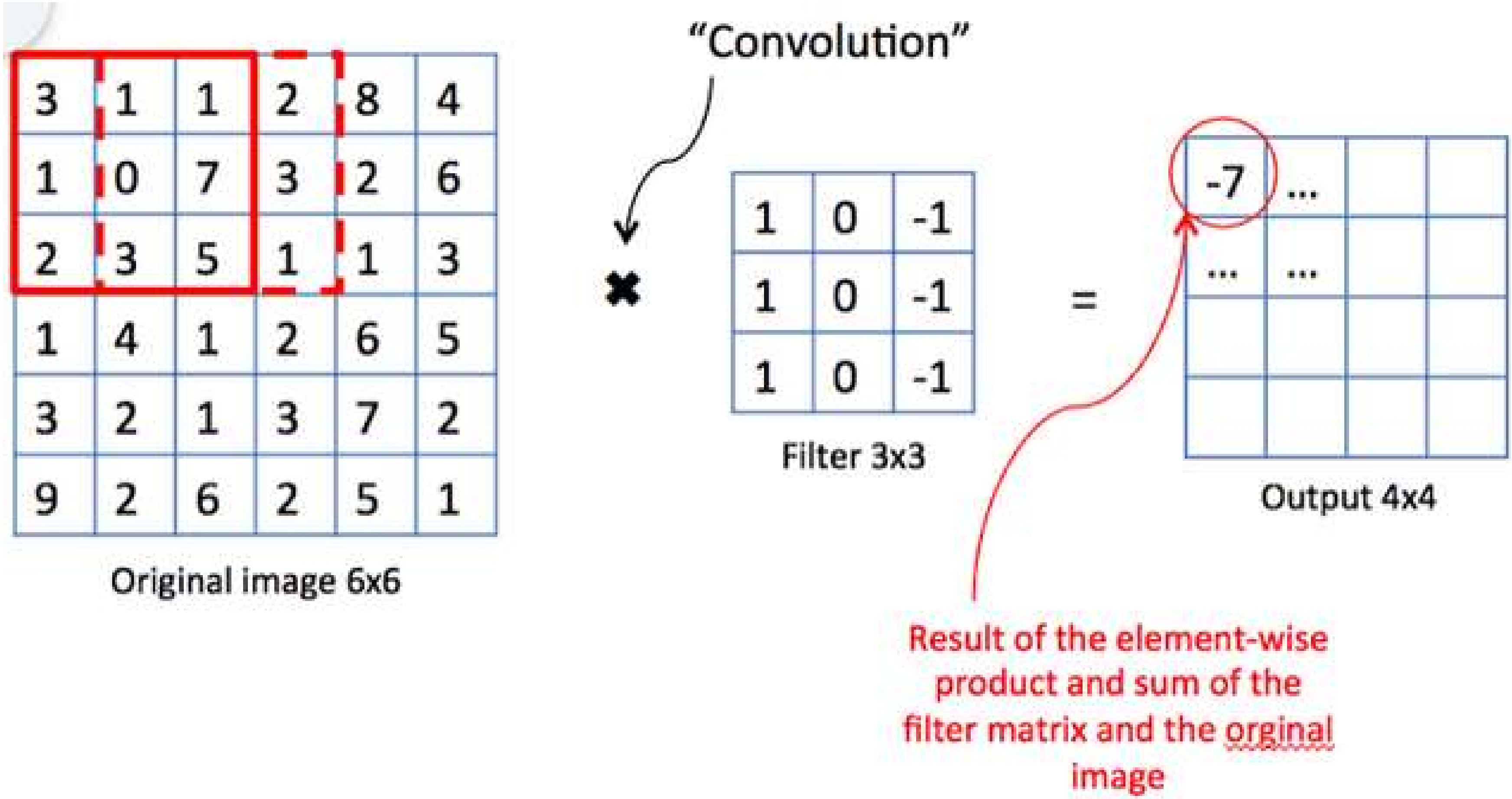
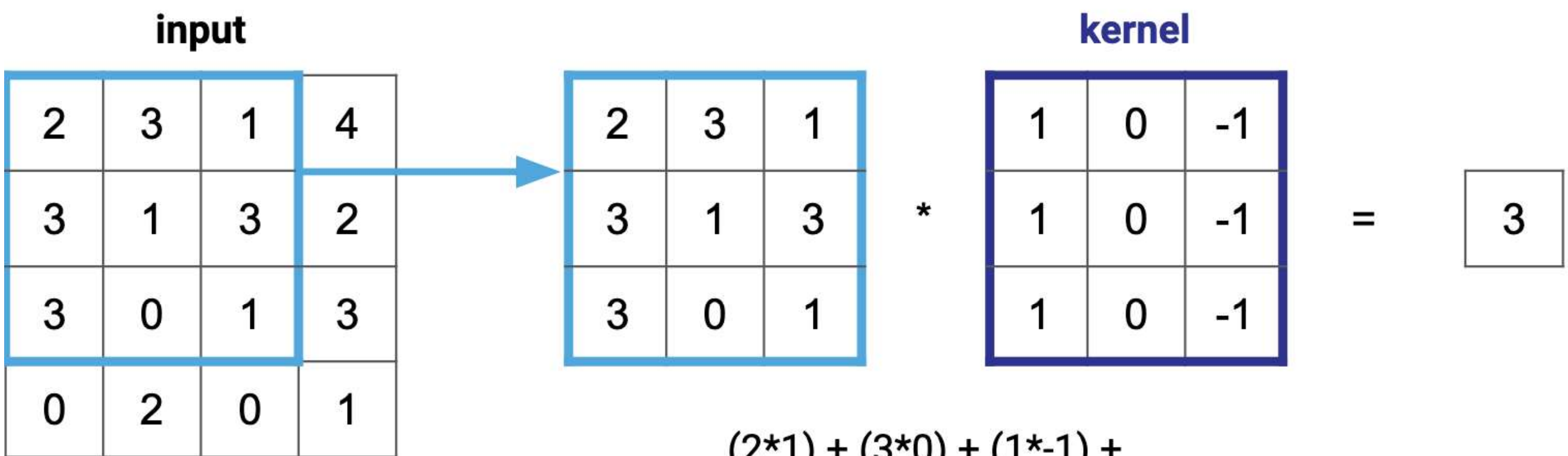


Image Convolution

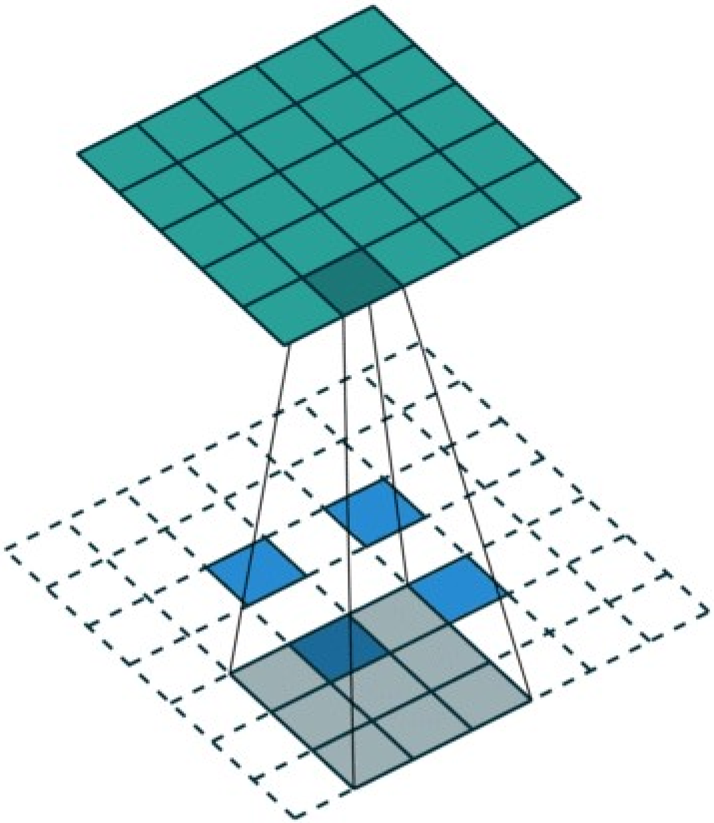
Image Convolution





$$(2*1) + (3*0) + (1*-1) +$$
$$(3*1) + (1*0) + (3*-1) +$$
$$(3*1) + (0*0) + (1*-1) = 3$$

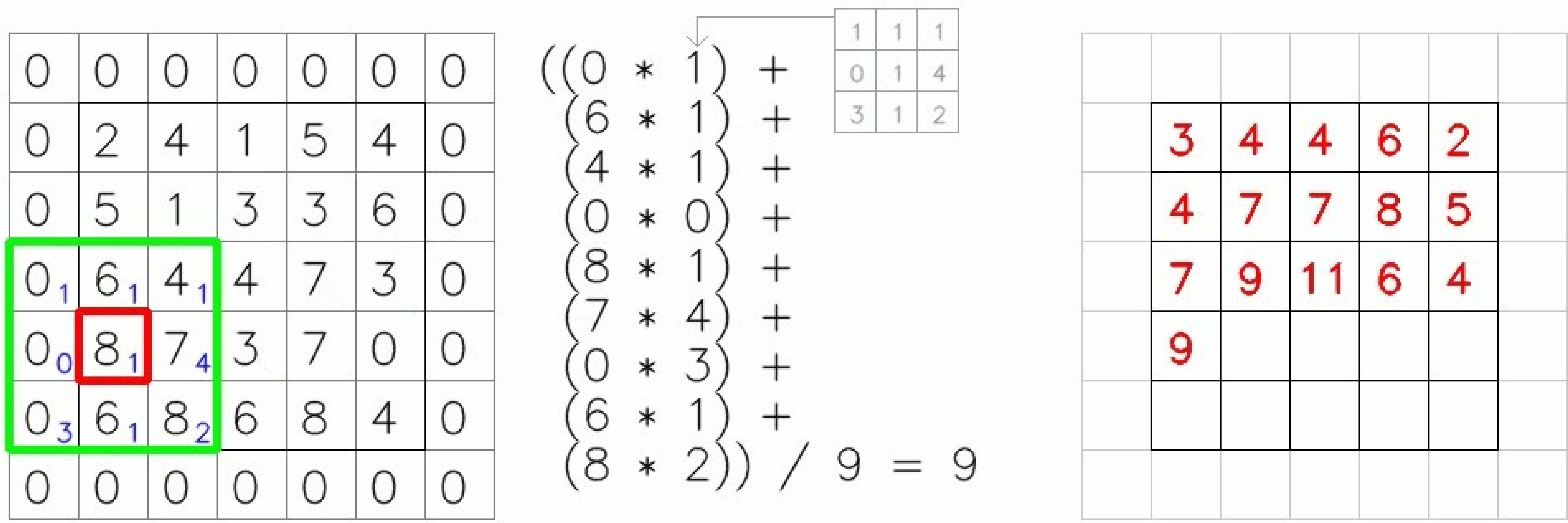
2D- Convolution



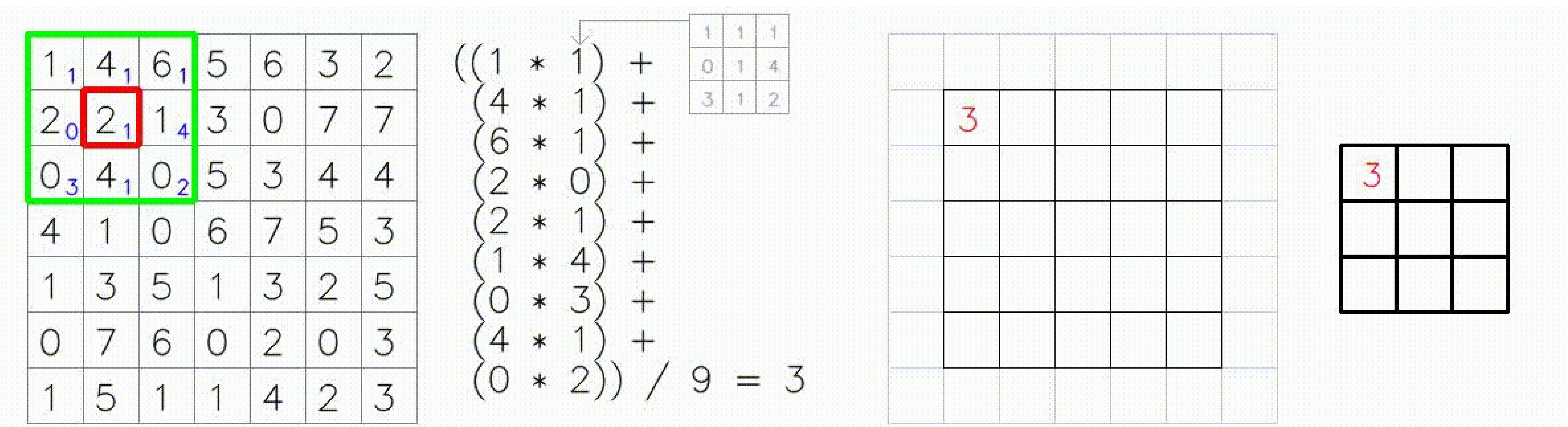
3	3	2	1	0
0	0	1_{0}	3_{1}	1_{2}
3	1	2_{2}	2_{2}	3_{0}
2	0	0_{0}	2_{1}	2_{2}
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

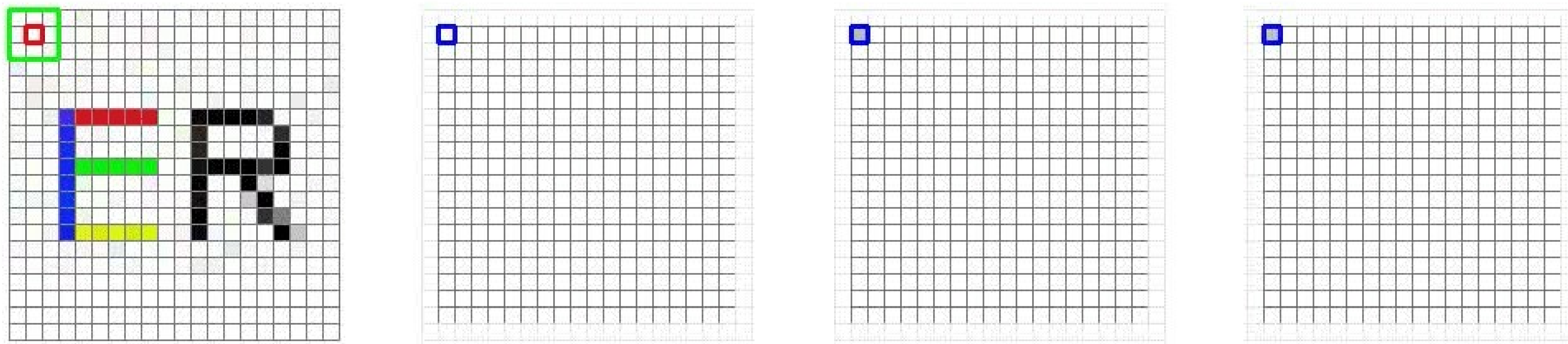
Padding



Padding & Stride



Blur & Sobel



1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Blur

1	2	1
0	0	0
-1	-2	-1

Sobel

Image Kernels

- [Image Kernels]()

0	-1	0
-1	5	-1
0	-1	0

sharpen

Blur

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

0	0	0
0	1	0
0	0	0

custom

-1	-2	-1
0	0	0
1	2	1

bottom sobel

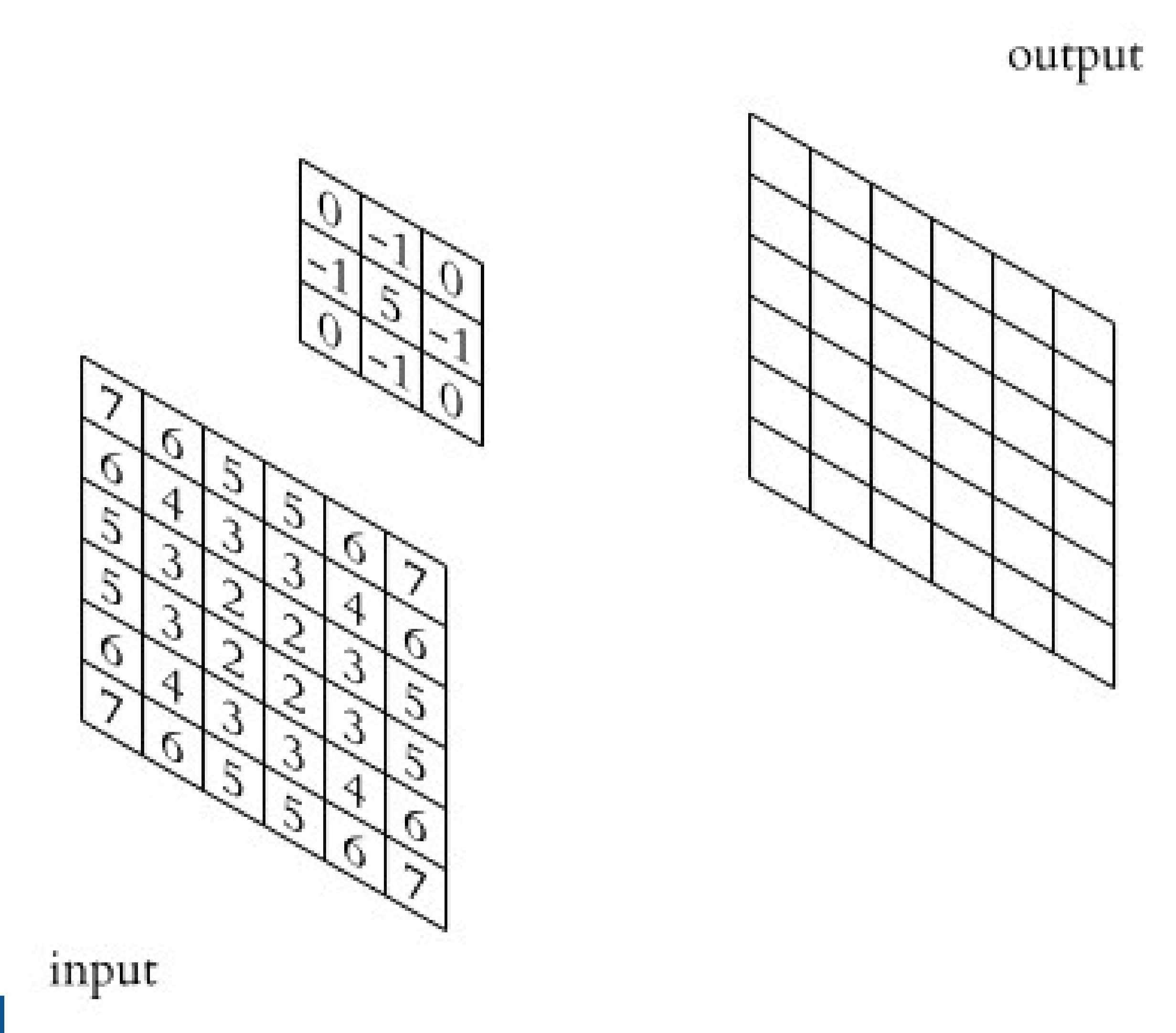
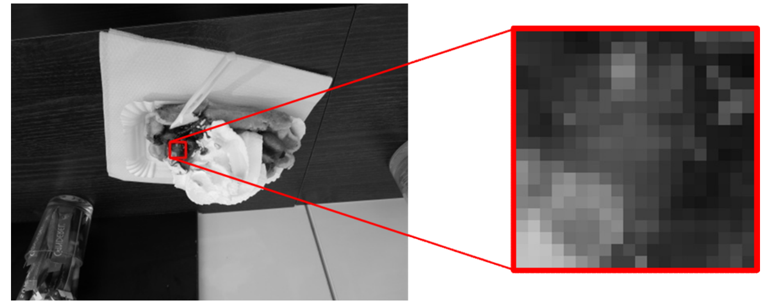
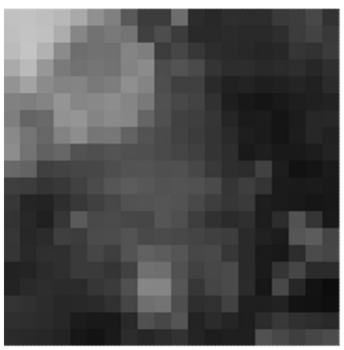

output
0 -1 0
-1 5 -1
0 -1 0
7 6 5 5 6 7
6 4 3 3 4 6
5 3 2 2 3 5
5 3 2 2 3 5
6 4 3 3 4 6
7 6 5 5 6 7
input

Image processing using Kernel





=

193	194	190	184	169	144	128	89	60	60	109	44	40	46	45	45	58	61	72	50
191	195	192	173	134	114	121	116	64	77	60	41	41	43	41	41	46	60	67	57
187	196	178	139	110	113	112	132	126	61	70	55	45	42	40	37	39	53	50	59
170	186	151	122	114	117	114	131	139	76	83	74	52	45	45	41	43	46	39	47
147	163	139	131	132	121	125	143	132	78	64	64	42	33	35	30	32	36	33	43
129	126	132	148	134	136	141	133	121	81	72	67	49	30	24	21	25	30	32	34
126	101	106	146	149	132	138	134	101	80	65	62	53	37	27	28	21	28	39	40
137	117	103	130	141	118	119	99	83	74	66	60	52	42	30	27	21	29	39	33
141	115	97	103	82	79	84	80	79	74	69	64	52	45	26	31	25	25	29	35
105	99	64	67	70	71	78	83	83	79	80	72	57	46	44	65	29	18	21	27
63	60	52	56	65	75	86	92	87	82	83	81	74	53	62	52	27	26	27	25
55	37	35	46	56	64	71	82	83	85	82	73	62	54	58	30	27	25	22	24
51	40	32	54	90	73	75	71	70	85	79	62	49	43	47	29	44	82	53	25
54	39	43	63	61	75	76	83	73	75	72	68	64	61	42	33	29	86	92	69
53	41	47	47	58	70	62	83	97	92	85	71	73	55	42	27	25	45	75	70
34	36	43	47	58	58	53	65	112	111	81	65	71	74	41	28	53	70	43	43
35	42	45	43	43	54	59	80	134	133	86	55	69	74	43	23	72	39	19	13
52	48	44	39	40	42	49	69	111	115	78	46	49	65	31	31	34	19	18	15
47	46	43	39	28	25	25	40	80	71	84	35	43	37	30	44	56	48	45	41
43	40	39	35	23	19	26	44	38	44	50	36	29	31	31	81	72	63	70	76

193	194	190	184	169	144	128	89	60	60	109	44	40	46	45	45	58	61	72	50
191	195	192	173	134	114	121	116	64	77	60	41	41	43	41	41	46	60	67	57
187	196	178	139	110	113	112	132	126	61	70	55	45	42	40	37	39	53	50	59
170	186	151	122	114	117	114	131	139	76	83	74	52	45	45	41	43	46	39	47
147	163	139	131	132	121	125	143	132	78	64	64	42	33	35	30	32	36	33	43
129	126	132	148	134	136	141	133	121	81	72	67	49	30	24	21	25	30	32	34
126	101	106	146	149	132	138	134	101	80	65	62	53	37	27	28	21	28	39	40
137	117	103	130	141	118	119	99	83	74	66	60	52	42	30	27	21	29	39	33
141	115	97	103	82	79	84	80	79	74	69	64	52	45	26	31	25	25	29	35
105	99	64	67	70	71	78	83	83	79	80	72	57	46	44	65	29	18	21	27
63	60	52	56	65	75	86	92	87	82	83	81	74	53	62	52	27	26	27	25
55	37	35	46	56	64	71	82	83	85	82	73	62	54	58	30	27	25	22	24
51	40	32	54	90	73	75	71	70	85	79	62	49	43	47	29	44	82	53	25
54	39	43	63	61	75	76	83	73	75	72	68	64	61	42	33	29	86	92	69
53	41	47	47	58	70	62	83	97	92	85	71	73	55	42	27	25	45	75	70
34	36	43	47	58	58	53	65	112	111	81	65	71	74	41	28	53	70	43	43
35	42	45	43	43	54	59	80	134	133	86	55	69	74	43	23	72	39	19	13
52	48	44	39	40	42	49	69	111	115	78	46	49	65	31	31	34	19	18	15
47	46	43	39	28	25	25	40	80	71	84	35	43	37	30	44	56	48	45	41
43	40	39	35	23	19	26	44	38	44	50	36	29	31	31	81	72	63	70	76

138	134	101
119	99	83
84	80	79

Sharpening Image

$$\begin{bmatrix} 138 & 134 & 101 \\ 119 & 99 & 83 \\ 84 & 80 & 79 \end{bmatrix} * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} & & \\ & 79 & \\ & & \end{bmatrix}$$

Writing this out: $(-1)(134) + (-1)(83) + (-1)(80) + (-1)(119) + (5)(99) = 79.$

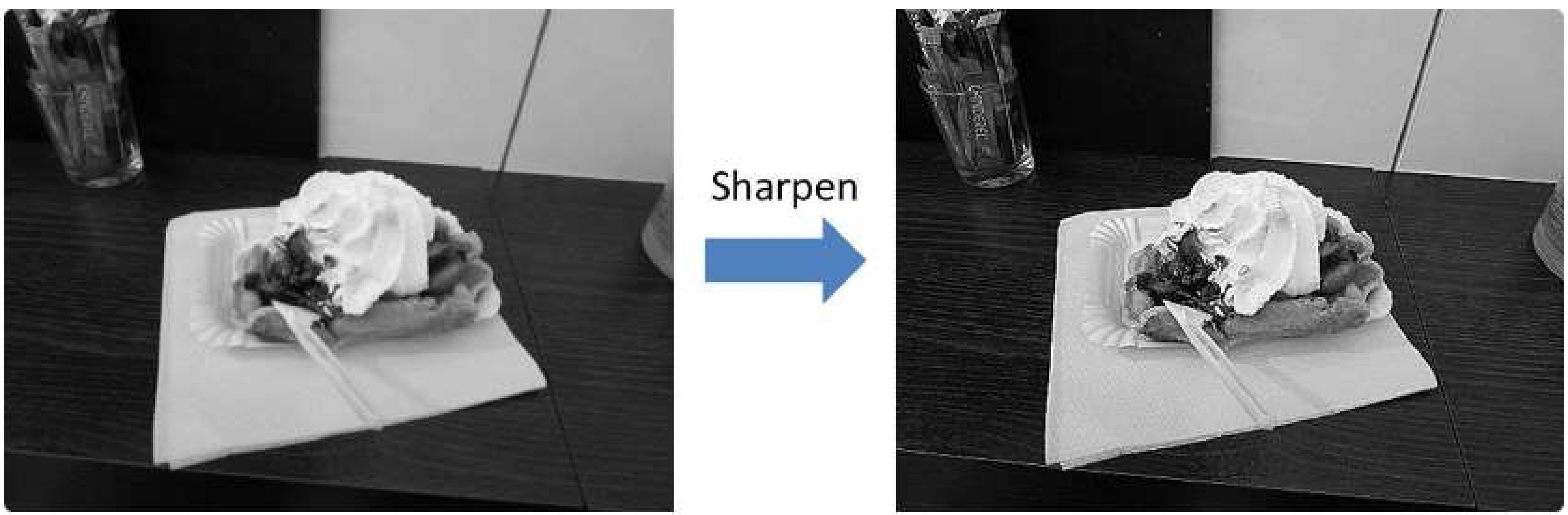
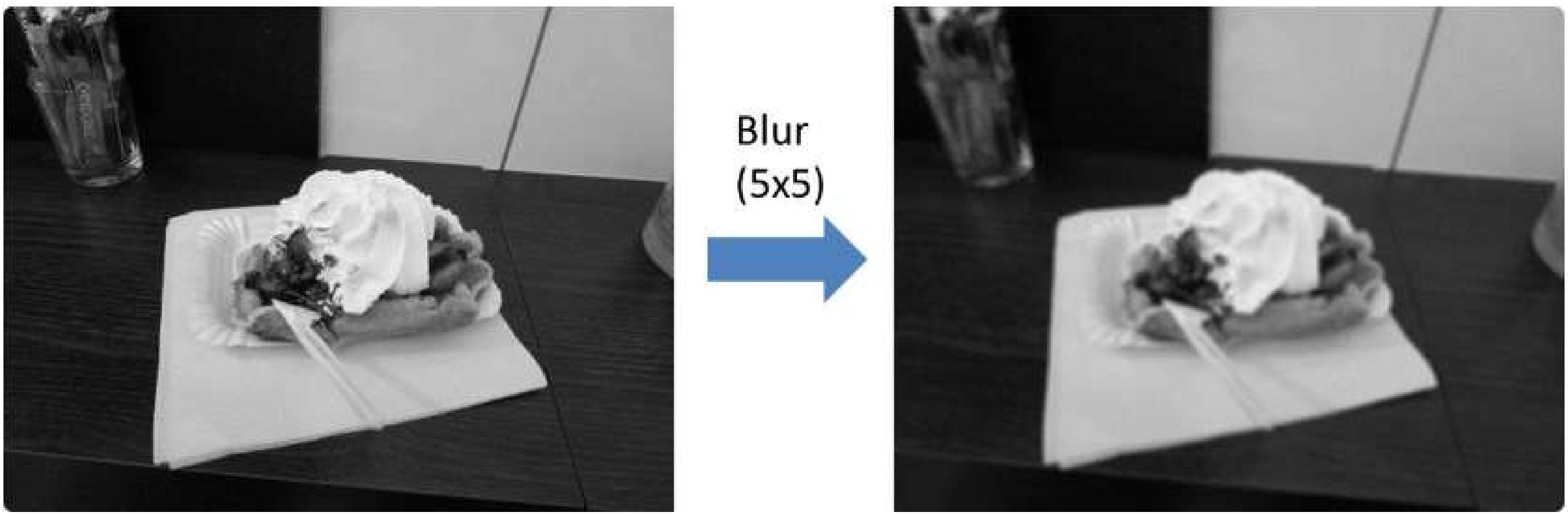


Image Blurring

$$\begin{bmatrix} 138 & 134 & 101 \\ 119 & 99 & 83 \\ 84 & 80 & 79 \end{bmatrix} * \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

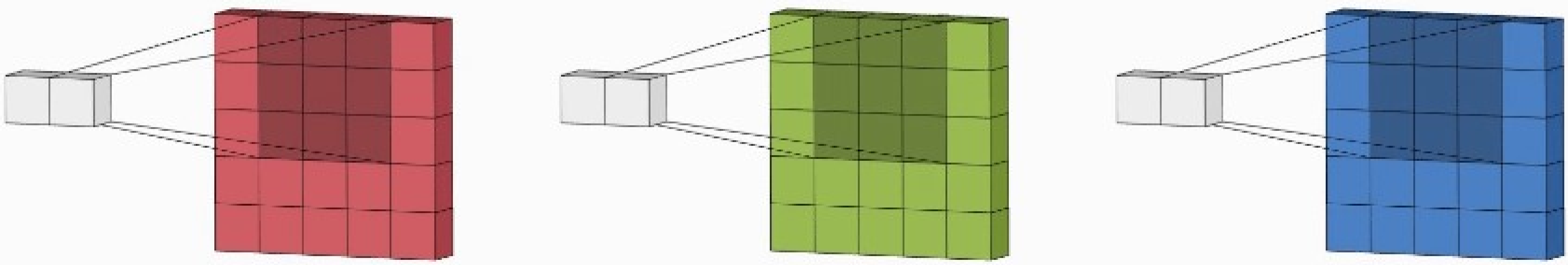


$$output\_size = ((input\_size - kernel\_size + 2 * padding) / stride) + 1$$

$$\overset{\text{Input}}{\begin{bmatrix}1&2&3&4\\5&6&7&8\\9&10&11&12\\13&14&15&16\end{bmatrix}} \star \overset{\text{Kernel}}{\begin{bmatrix}0&-1&0\\-1&5&-1\\0&-1&0\end{bmatrix}} + \overset{\text{Bias}}{\begin{bmatrix}5\end{bmatrix}} = \overset{\text{Output}}{\begin{bmatrix}11&12\\15&16\end{bmatrix}}$$

★ Cross-Correlation

2D Convolution for Multichannel Image



2D Convolution for Multichannel Image

Input

1	0	1	0	2
1	1	3	2	1
1	1	0	1	1
2	3	2	1	3
0	2	0	1	0

1	0	0	1	0
2	0	1	2	0
3	1	1	3	0
0	3	0	3	2
1	0	3	2	1

2	0	1	2	1
3	3	1	3	2
2	1	1	1	0
3	1	3	2	0
1	1	2	1	1

Kernel

0	1	0
0	0	2
0	1	0

2	1	0
0	0	0
0	3	0

1	0	0
1	0	0
0	0	2

Intermediate Output

7	5	3
4	7	5
7	2	8

5	3	10
13	1	13
7	12	11

7	5	2
11	8	2
9	4	6

Output

19	13	15
28	16	20
23	18	25

Convolution of a 3 channel image with a 3x3x3 kernel

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

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

Kernel Channel #1

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

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

Kernel Channel #3

$$308 + -498 + 164 + 1 = -25$$

Bias = 1

Output

-25				...
				...
				...
				...
...	...	...	...	...

Noise in Image

Random variations of brightness or color information in images, typically caused by sensor or electronic interference during image acquisition, transmission, or storage. Noise can obscure important details and degrade the quality of an image, making it more challenging to analyze or process.

Effects of Noise

- **Degradation of Image Quality**: Reduced clarity and visibility of important details.
- **Distortion of Features**: Critical features might be obscured or distorted.
- **Increased Difficulty in Analysis**: More challenging to perform accurate image analysis and interpretation.

Types of Noise

- **Salt-and-Pepper Noise (Impulse Noise)**: Random occurrences of black and white pixels, often due to errors in data transmission.
- **Poisson Noise (Shot Noise)**: Arises from the statistical nature of photon detection in imaging sensors
- **Gaussian Noise**: Random variations in intensity values following a Gaussian distribution.
- **Speckle Noise**: Multiplicative noise common in radar and medical imaging.
- .

From left to right: Salt and pepper (Wikimedia), Shot (Mdf), Gaussian (Anton), Inference (Max221), Film Grain (bottom right. Tristan Bowersox)

Noise Models

1. Additive Noise Model

$$I_{\text{noisy}}(x, y) = I_{\text{original}}(x, y) + N(x, y)$$

- Description: Noise $N(x, y)$ is added to the original image $I_{\text{original}}(x, y)$.
- Example: Gaussian noise.

2. Multiplicative Noise Model

$$I_{\text{noisy}}(x, y) = I_{\text{original}}(x, y) \cdot N(x, y)$$

- Description: Noise $N(x, y)$ multiplies the original image $I_{\text{original}}(x, y)$.
- Example: Speckle noise.

Noise Removal Techniques

- **Spatial Domain Filtering**
  - **Mean Filter**: Averages the pixel values in a neighborhood.
  - **Median Filter**: Replaces each pixel value with the median of the surrounding pixel values.
  - **Gaussian Filter**: Applies a Gaussian function to weight neighboring pixels.
- **Frequency Domain Filtering**
  - **Fourier Transform**: Transforms the image to the frequency domain for filtering.
  - **Wiener Filter**: Combines inverse filtering with noise

Image Filtering

- Filtering techniques in image processing are used to enhance or extract information from images.
- Linear Filters:
  - Averaging Filter (Mean Filter)
  - **Gaussian Filter**
  - Sobel Filter
  - Laplacian Filter
- Non-Linear Filters
  - Median Filter
  - Bilateral Filter
- Advanced Filtering Techniques
  - Non-Local Means Denoising
  - Gabor Filter

Linear Filter

Averaging Filter (Mean Filter)

- **Purpose**: Smooths the image by reducing noise.
- **Operation**: Computes the average of the pixel values in the neighborhood and assigns it to the central pixel.

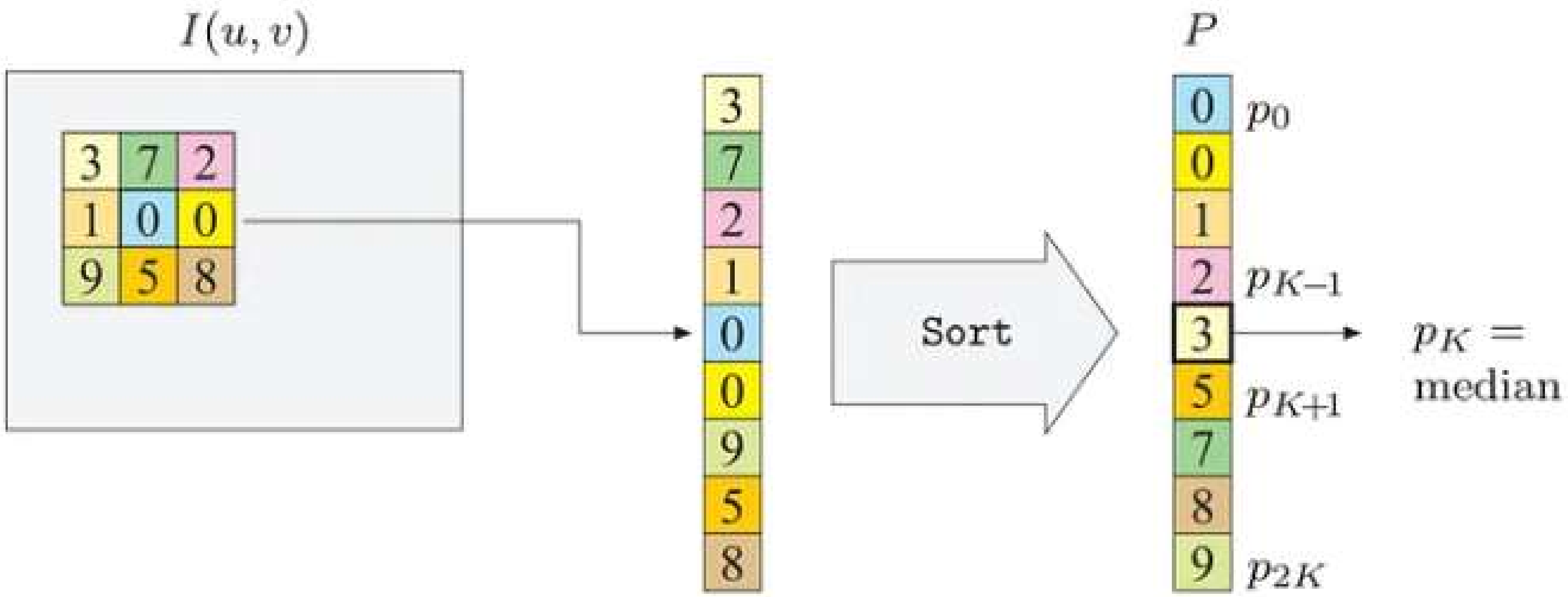
3	9	11	2
7	15*	8	8
10	12	9	10
1	9	11	2

New Pixel : (3+9+11+7+15+8+10+12+9)/9
=9

Non-Linear Filter

Median Filter

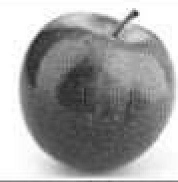
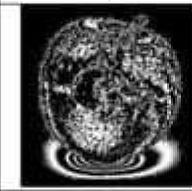
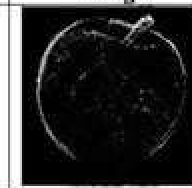
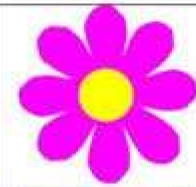
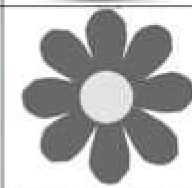
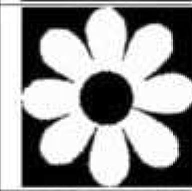
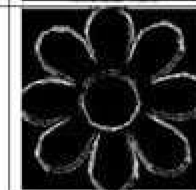
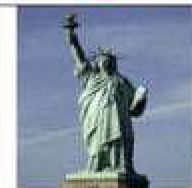
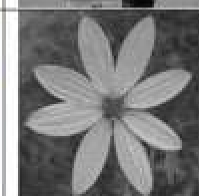
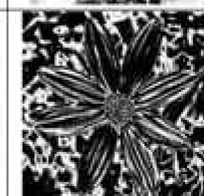
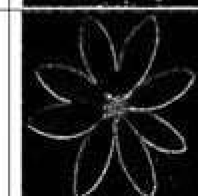
- **Purpose**: Reduces noise while preserving edges.
- **Operation**: Replaces each pixel with the median of the pixel values in the neighborhood.



Advanced Filtering Techniques

Gabor Filter

- **Purpose**: Detects specific frequency and orientation components in the image.
- **Operation**: Uses a sinusoidal wave modulated by a Gaussian envelope.

Sr.No	Image name	Input image	Gray image	Gabor filtered Image	Gabor Filter Edge Detected image
1	Apple (256x256)				
2	Flower (256x256)				
3	Statue of Liberty (512x512)				
4	Sun Flower (512x512)				
5	Lena (512x512)