Linear Algebra and Calculus for Computer Science Engineering

25CS201

Competency: ***Use*** the principles of Linear Algebra and Calculus to solve problems in Computer Science Engineering.

LU – I Matrices

CO1: ***Use*** matrix operations, eigenvalues, and eigenvectors to enhance data efficiency.

PrO-1. ***Convert the given*** JPG image into a matrix and enhance its brightness and contrast using matrix operation

LUO-1a ***<u>Find</u>*** the principal components for ***<u>the given</u>*** dataset in data compression using eigenvalues and eigenvectors in Principal Component Analysis (PCA).

SLO-1:

- ***<u>Represent</u>*** the pixel values of a JPG image in matrix form for further computational processing

SLO-2:

- ***<u>Enhance</u>*** image brightness and contrast using matrix addition and scalar multiplication.

Job Role	Company	Required Skills	Estimated Salary (INR)
Data Scientist	Google, TCS, Mu Sigma, Accenture	Python, NumPy, Pandas, Matplotlib, SQL, Machine Learning, **Linear Algebra**	₹6–15 LPA
Machine Learning Engineer	Amazon, Infosys, Fractal Analytics, Microsoft	Python, Scikit-learn, TensorFlow, Deep Learning, **Linear Algebra**	₹8–20 LPA
Computer Vision Engineer	Nvidia, Cognizant, Wipro, Meesho	OpenCV, CNNs, Image Processing, Python, PyTorch, **Matrix Algebra**	₹6–18 LPA
Deep Learning Engineer	OpenAI, Zoho, Flipkart, Samsung R&D	PyTorch, TensorFlow, Neural Networks, GPU Computing, **Matrix Operations**	₹8–22 LPA
Software Developer (Scientific)	ISRO, DRDO, MathWorks, IBM	MATLAB, Python, Numerical Computing, Algorithms, **Linear Algebra**	₹5–12 LPA

Game Developer / Graphics Programmer	**Ubisoft, EA Games, Unity Technologies, BYJU'S**	**C++, OpenGL, Unity, 3D Transformations, Shader Programming, Matrix Math**	**₹4–12 LPA**
Robotics Engineer	ABB, Tata Elxsi, Boston Dynamics, IITs	ROS, Kinematics, Python, Control Systems, **Matrix Transformations**	₹6–14 LPA
Cryptography Specialist	DRDO, Infosys, Mastercard, Qualcomm	Modular Arithmetic, **Matrix Algebra**, Cryptography, Python, C++	₹7–16 LPA
Research Scientist	Google Research, Adobe Labs, IITs, IISc	MATLAB, **Matrix Theory**, Optimization, Python, Algorithm Design	₹8–18 LPA
CAD/CAE Developer	Siemens, Dassault Systèmes, Autodesk, L&T Tech	C++, Finite Element Analysis, MATLAB, CAD Tools, **Matrix Computation**	₹6–14 LPA

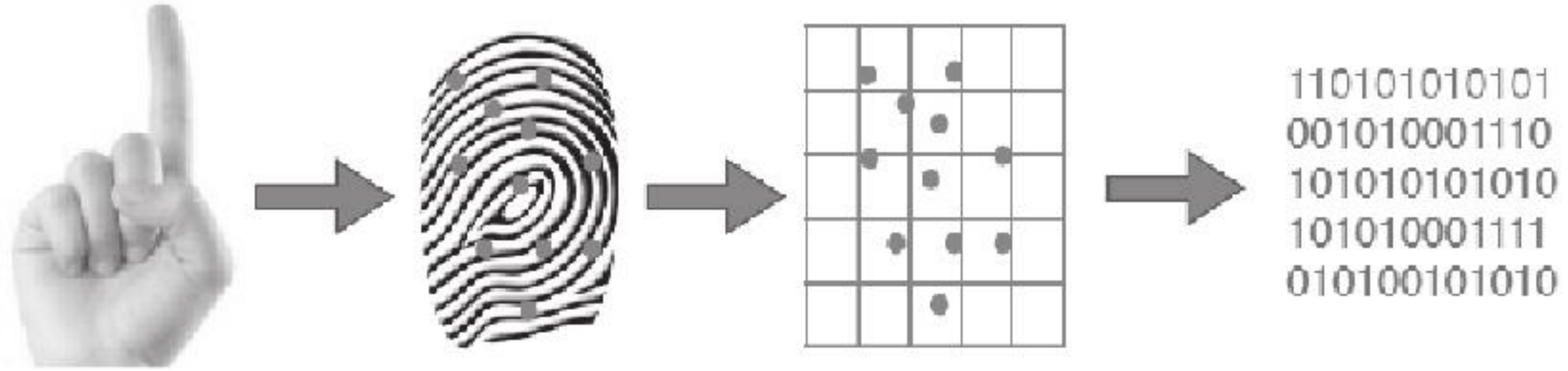
Biometric Systems

Karunya

Purpose of Biometrics

- **Security & Authentication:** Used in devices (like phones), airports, border control, and banks for verifying identity.
- **Access Control:** Only authorized persons can access data or physical spaces.
- **Surveillance & Forensics:** Identifying suspects or verifying identity in crime investigations.

How it works...



Capture the biometric input:

•A **fingerprint scanner** captures the fingerprint image.

•This could be on a phone, biometric device, or scanner.

Feature Extraction:

•The fingerprint image is processed to **identify key features** (like ridges, minutiae, points of bifurcation, etc.).

•These key points are the distinguishing features of the fingerprint.

Matrix Mapping:

•These features are then **mapped into a matrix/grid**.

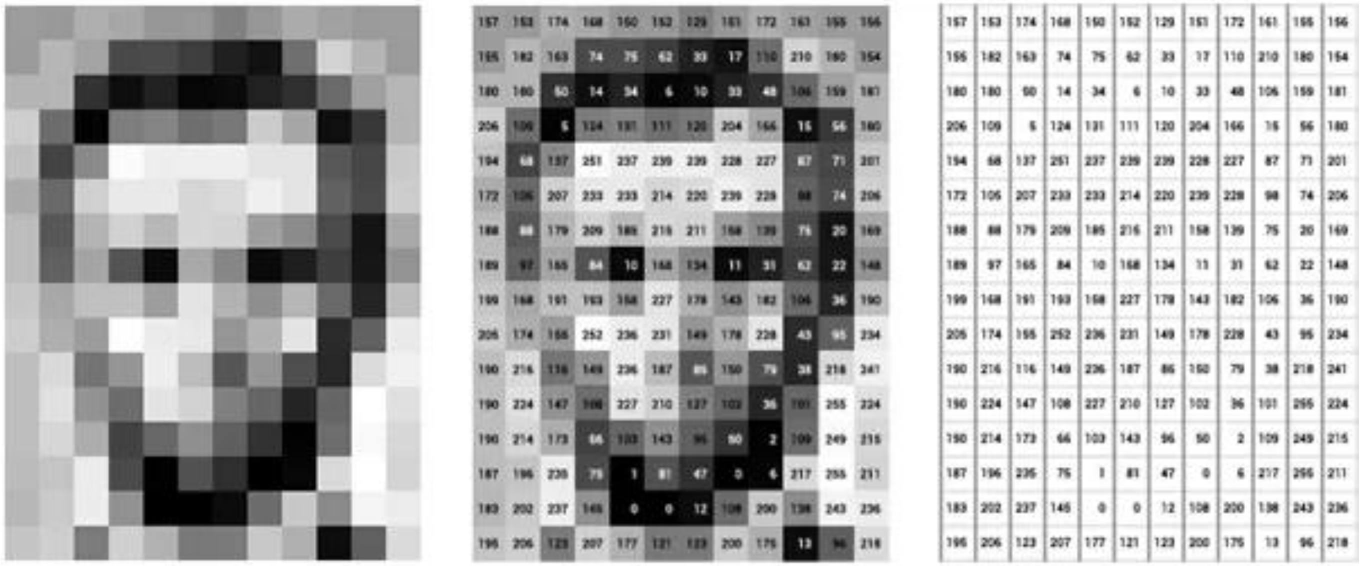
•Each cell in the grid corresponds to a **spatial coordinate**, storing whether a feature exists at that position.

Binary/Encoded Representation:

•The matrix data is **converted into binary or numerical codes**.

•This allows the system to **store**, **compare**, and **match** fingerprints efficiently

How it works...



Face Recognition Using Matrices

1.Image to Pixels :

1. A face image is captured (example: grayscale image).
2. It's shown as a **pixelated version**, where each square is a **pixel**.

2.Pixel Intensity Matrix :

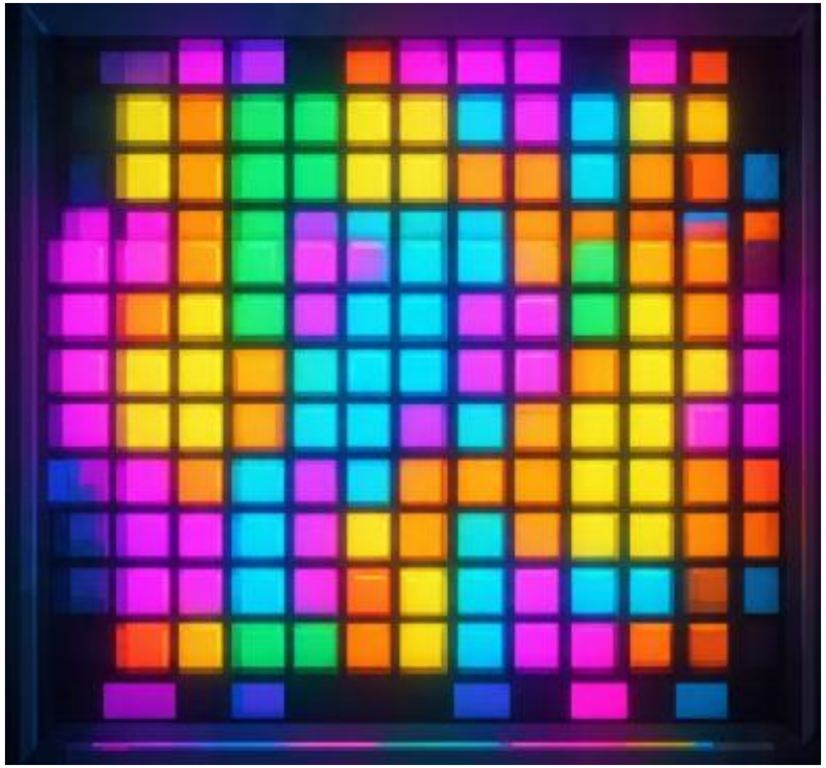
1. The image is converted to a **matrix of intensity values** (grayscale from 0–255).
2. Each cell in the matrix represents how dark or light the corresponding pixel is.

3.Numerical Matrix:

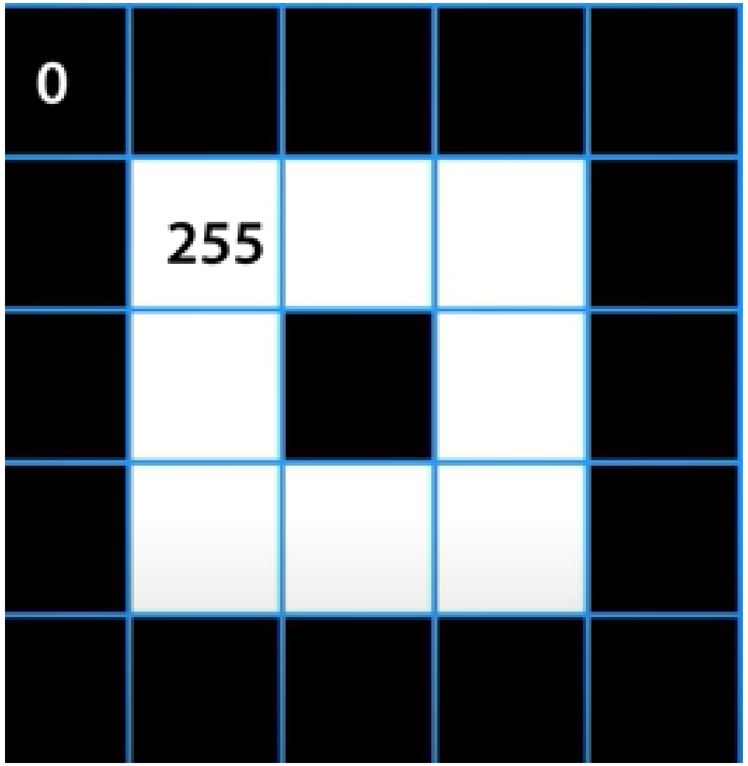
1. You now have a **numerical matrix** that can be used for further processing.
2. This matrix is input to **face recognition algorithms** like PCA, Eigenfaces, or deep learning models.

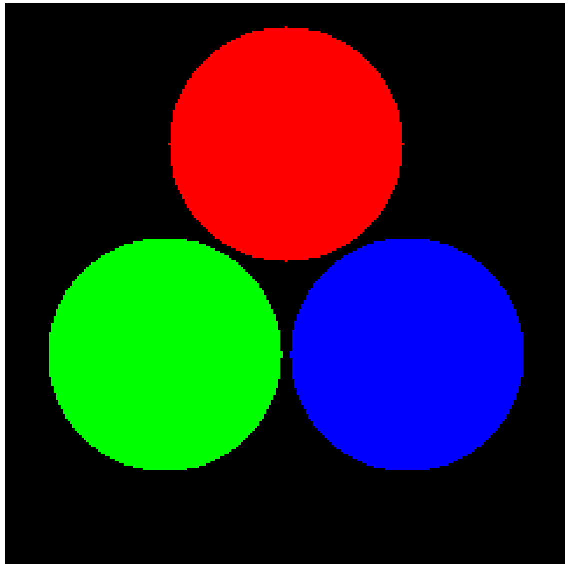
Biometric Type	Matrix Usage
Fingerprint	**Binary matrix** of minutiae positions and ridge patterns
Face Recognition	**Matrix of pixel intensities** (grayscale or RGB)
Iris/Retina Scan	Polar coordinate-based **matrix of intensity patterns**
Palm Veins	Infrared image transformed into **matrix for vein pattern extraction**

PIXEL MATRIX




binary image
greyscale image
RGB image





- Each inner [R, G, B] triplet is a **color pixel**.
- [255, 0, 0] = Red
- [0, 255, 0] = Green
- [0, 0, 255] = Blue
- [255, 255, 0] = Yellow

(120, 45, 200)	(34, 210, 90)	(180, 30, 255)	(200, 150, 20)
(80, 190, 220)	(100, 50, 90)	(255, 255, 0)	(30, 90, 200)
(60, 80, 30)	(140, 250, 100)	(90, 10, 190)	(200, 180, 50)
(30, 60, 180)	(220, 130, 10)	(150, 200, 250)	(10, 240, 150)
(90, 40, 200)	(200, 180, 120)	(50, 255, 100)	(180, 10, 220)
(90, 40, 200)	(200, 180, 120)	(50, 255, 100)	(180, 10, 220)

PIXEL MATRIX

- A pixel matrix (also called an image matrix) is a 2D (or 3D) array of numbers where each element represents the intensity or color value of a pixel in an image.
- Each row of the matrix corresponds to a horizontal line of pixels.
- Each column represents a vertical line of pixels.
- The value of each element depends on the image type:

  - Grayscale: A single intensity value (0–255).
  - Color (RGB): A tuple/list of 3 values (R, G, B) for each pixel.
  - Binary: Values are 0 (black) or 1 (white).

Types of Pixel Matrices:

Image Type	Matrix Dimensions	Example Matrix Content
Grayscale Image	M x N	Single intensity value (0–255)
RGB Color Image	M x N x 3	Red, Green, Blue values per pixel
Binary Image	M x N	Only 0 (black) or 1 (white)

PIXEL MATRIX

Types of 8-bit Images:

8 bits = 1 byte = 2^8 = 256 possible values

1. Grayscale 8-bit Image:
Each pixel has 256 possible shades of gray, ranging from:
0 (black) to 255 (white).
Common in medical imaging, document scanning, etc.

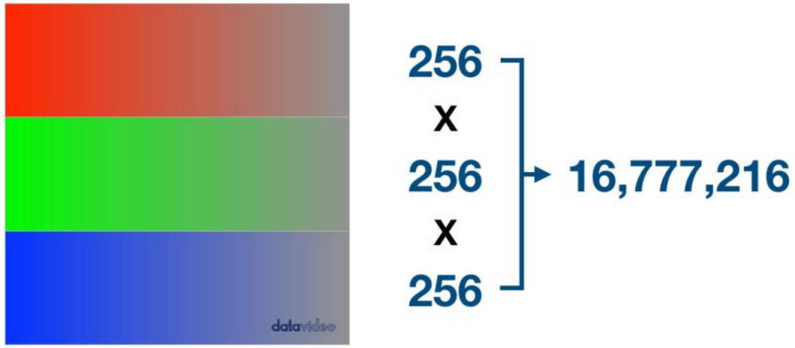
230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

2. 8-bit Indexed Color Image (Palette-based):
Each pixel stores 1 byte (8 bits), which points to a color in the palette of 256 predefined colors.

Animations, icons, Web graphics, Basic image storage

PIXEL MATRIX

A **True Color RGB Image** is a digital image format where each pixel directly stores **full Red, Green, and Blue (RGB) color values**, typically using **8 bits per channel**.



3 x 8-bit channels

8 bits for R, 8 for G, 8 for B

Bits per Pixel (bpp)	Possible Values	Image Type
1-bit	2 (black or white)	Binary images
4-bit	16	Early color images
8-bit	256	Grayscale or single-channel color
24-bit (8×3)	16.7 million	True color (RGB)

PIXEL MATRIX

⬜ Grayscale Image Matrix:

- A **grayscale image** contains **intensity values only** (shades of gray).
- Represented as a **2D matrix** (rows × columns), where each entry is a **single number** (pixel intensity).

📌 Example (3×3 grayscale):

$$\begin{bmatrix} 50 & 100 & 150 \\ 60 & 110 & 160 \\ 70 & 120 & 170 \end{bmatrix}$$

- Each value is a number between **0 (black)** and **255 (white)** in 8-bit images.

PIXEL MATRIX

🌈 Color Image Matrix:

- A **color image** is composed of **three channels: Red (R), Green (G), Blue (B)**.
- Represented as a **3D matrix**: (rows × columns × 3).
- Each pixel has a **tuple of 3 values** (R, G, B).

📌 **Example (3×3 color image):**

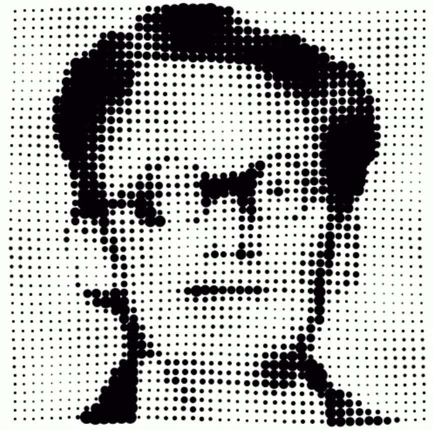
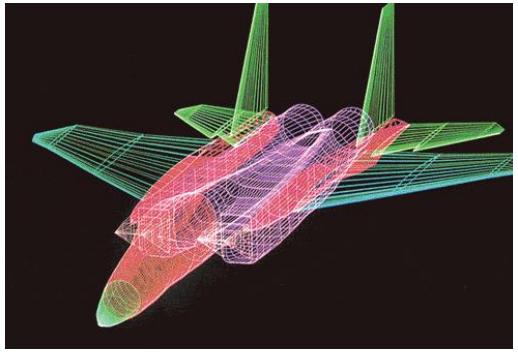
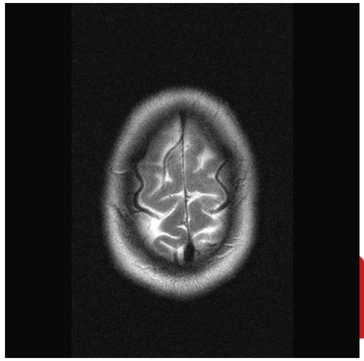
$$\begin{bmatrix} [(255,0,0), & (0,255,0), & (0,0,255)] \\ [(100,100,100), & (150,150,150), & (200,200,200)] \\ [(123,50,200), & (90,180,70), & (160,160,10)] \end{bmatrix}$$

- Each channel (R, G, B) has its own **2D matrix.**
- All three channels stacked = 3D matrix.

Karunya

🛠 Where is Pixel Matrix Used?

Field	Use of Pixel Matrix
Computer Vision	Object detection, segmentation, classification
Image Processing	Filtering, edge detection, brightness/contrast changes
Graphics Programming	Rendering, shading, transformations
Medical Imaging	CT/MRI scan interpretation
Machine Learning	Input for convolutional neural networks (CNNs)

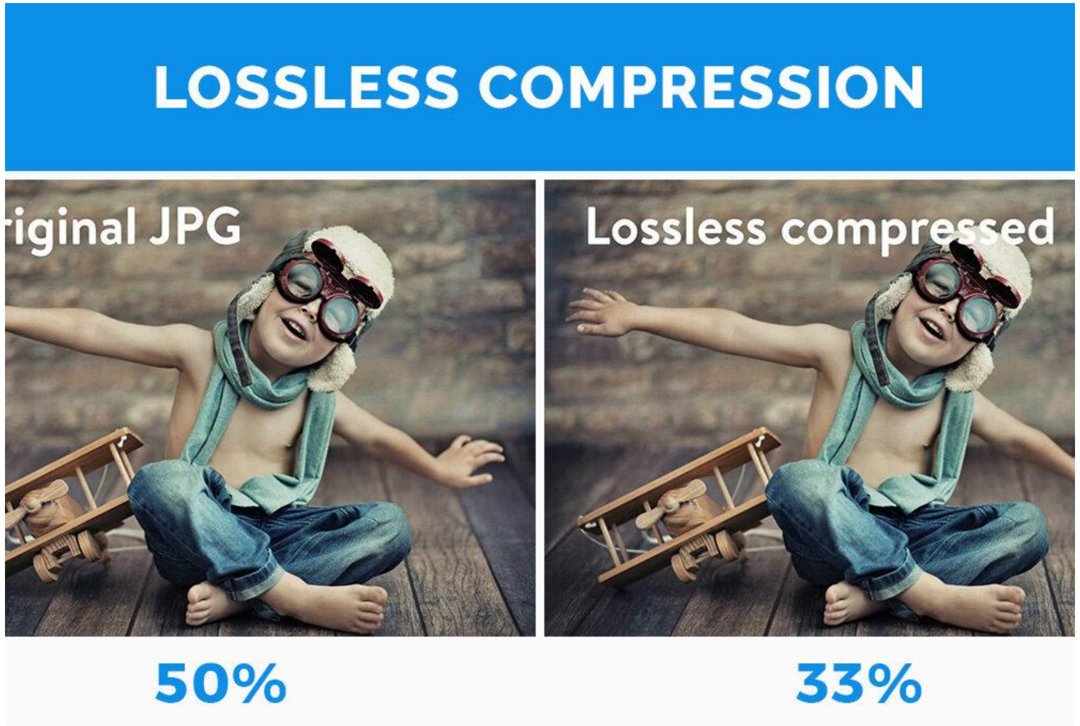





Importance of Pixel matrix

1. Digital Representation of Images

- Computers cannot see images like humans
- An image is stored as a matrix of pixel values (grayscale or RGB).

For image manipulation, filtering, or compression, computer algorithms operate on this pixel matrix.



Importance of Pixel matrix

2. Computer Vision & AI (Face Recognition, Object Detection)

- AI models (CNNs) take image matrices as input.
- Every pixel value contributes to how the model "learns" patterns in images.

Example: **Self-driving cars process video frames (image matrices) to detect lanes, traffic signals, etc.**

3. Graphics and Visualization

- Rendering images, animations, and 3D graphics involves heavy use of pixel matrices.
- Computer Graphics pipelines (OpenGL, DirectX) manipulate pixel data at matrix level.

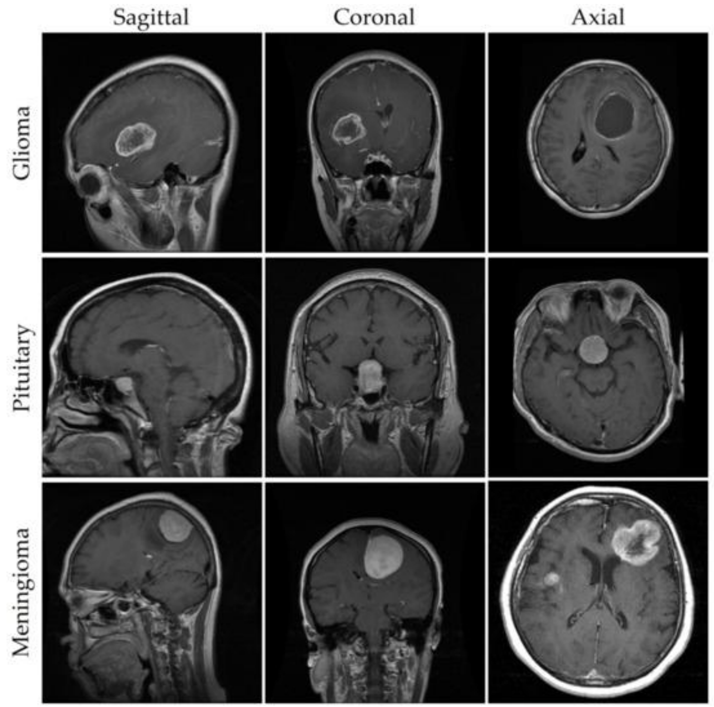
PIXEL MATRIX

4. Image Compression Algorithms (JPEG, PNG)

- Compression techniques like JPEG apply Discrete Cosine Transform (DCT) on image matrices.
- Efficient storage/transmission is achieved by reducing redundancies in the pixel matrix.

5. Medical Imaging & Remote Sensing

- CT scans, X-rays, Satellite images are processed by analyzing pixel intensity matrices.
- Algorithms detect tumors, land patterns, etc., using pixel matrix computations.



Matrix addition for brightness adjustment

Given Original Image Matrix:

$$\begin{bmatrix} 100 & 110 & 120 \\ 130 & 140 & 150 \\ 160 & 170 & 180 \end{bmatrix}$$

Result after Adding +40 Brightness:

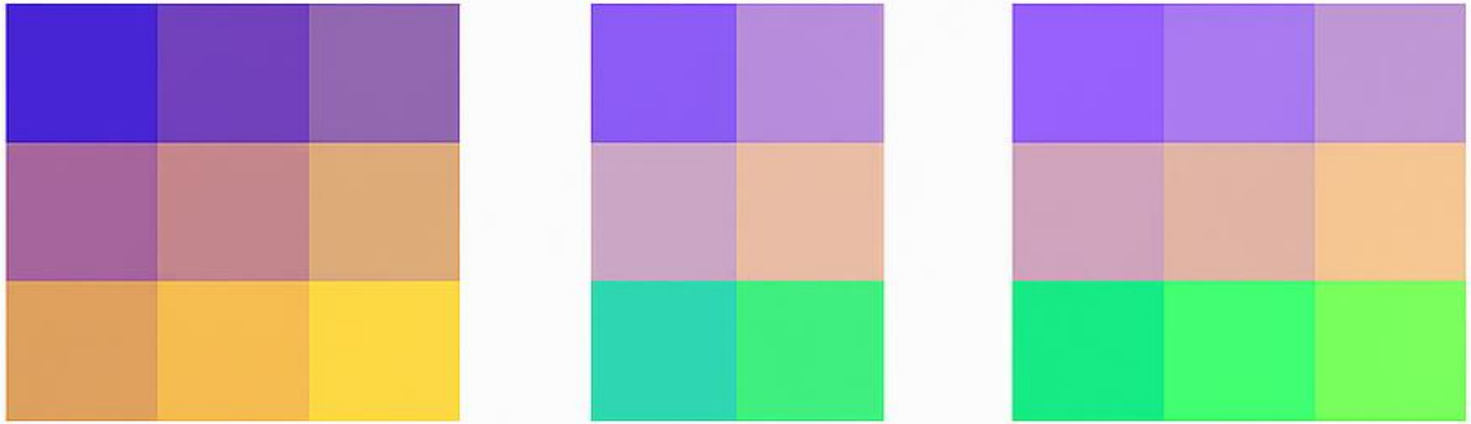
$$\begin{bmatrix} 140 & 150 & 160 \\ 170 & 180 & 190 \\ 200 & 210 & 220 \end{bmatrix}$$

Image appears brighter but retains details.

Karunya

Add brightness +40:

$$\begin{bmatrix} 100 & 110 & 120 \\ 130 & 140 & 150 \\ 160 & 170 & 180 \end{bmatrix} + \begin{bmatrix} 40 & 40 \\ 40 & 40 \\ 40 & 40 \end{bmatrix} = \begin{bmatrix} 140 & 150 & 160 \\ 170 & 180 & 190 \\ 200 & 210 & 220 \end{bmatrix}$$



Karunya

Matrix Multiplication for Contrast Adjustment (Scaling Pixel Values)

In image processing, contrast adjustment is often done by scaling the pixel values with a scalar multiplication. The contrast scaling factor (often denoted as **α**) controls the **amplitude of pixel intensity variations**

Formula:

New Pixel=α×Old Pixel

Where:

α = contrast scaling factor (>1 increases contrast, <1 decreases contrast)

α > 1: Increases contrast (dark gets darker, bright gets brighter).
α < 1: Decreases contrast (image appears flatter).
α = 1: No change.

Given a **Grayscale Image Matrix:**

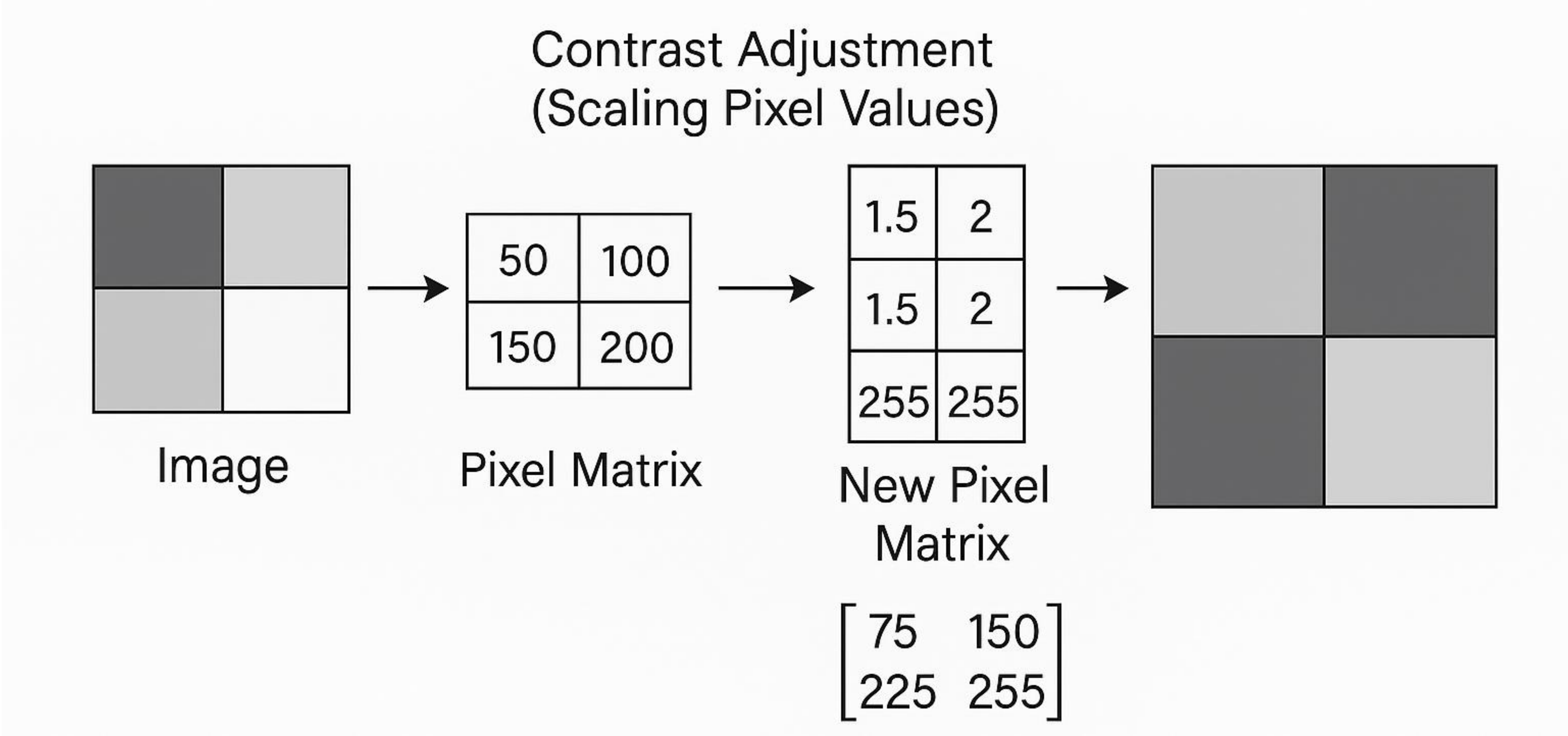
$$\begin{bmatrix} 50 & 100 \\ 150 & 200 \end{bmatrix}$$

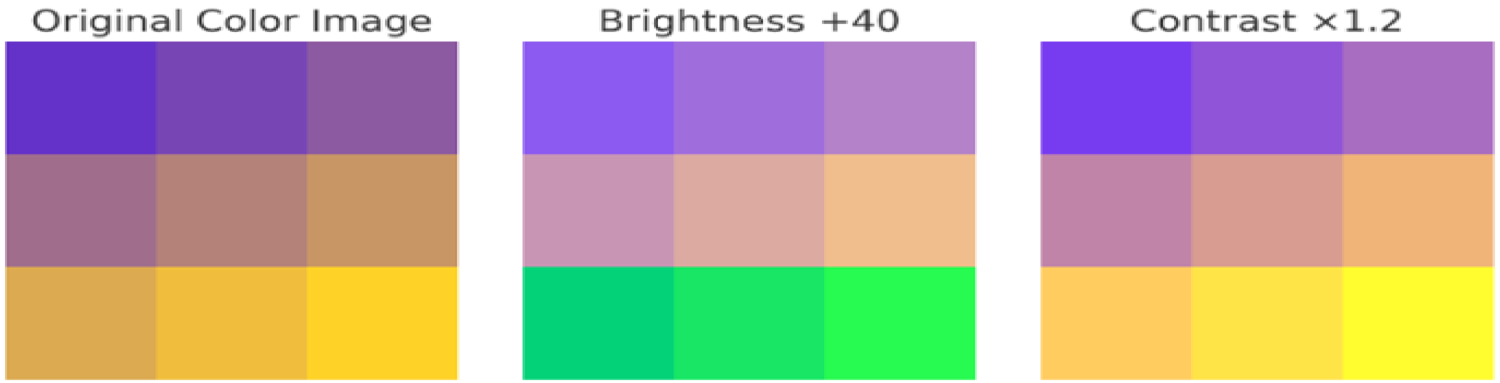
If you want to **increase contrast by 1.5x:**

$$\text{New Matrix} = 1.5 \times \begin{bmatrix} 50 & 100 \\ 150 & 200 \end{bmatrix} = \begin{bmatrix} 75 & 150 \\ 225 & 300 \end{bmatrix}$$

Then, clip values to 255 (for 8-bit images):

$$\begin{bmatrix} 75 & 150 \\ 225 & 255 \end{bmatrix}$$





Here's a visualization of **contrast and brightness enhancement** on a **color image**:

Real-Time Applications: Adding a Brightness Matrix (Brightness Enhancement)

- **Surveillance Cameras at Night:** Boosting brightness for low-light video feeds so security staff can detect objects more clearly.
- **Medical Imaging (X-rays, MRI):** Adjusting brightness so radiologists can spot faint structures or soft tissues.
- **Satellite Imaging:** Enhancing dark regions in images of Earth taken during dawn/dusk to improve visibility of terrain.
- **Smartphone Camera "Night Mode":** Brightness adjustment to make faces and objects more visible in dim light.

Real-Time Applications: Scalar Multiplication (Contrast Adjustment)

- **Facial Recognition in Varying Lighting:** Adjusting contrast so algorithms detect facial features in glare or shadow.
- **Industrial Machine Vision:** Improving contrast so defects on products (scratches, cracks) stand out in manufacturing inspection systems.
- **Astronomy Image Processing:** Enhancing contrast to reveal faint stars or galaxy structures against a dark background.
- **License Plate Recognition at Night:** Making reflective numbers more distinct from the dark surroundings.

Medical & Healthcare

- **MRI / CT / X-Ray Imaging:**
  Brightness adjustment highlights low-density tissue, and contrast enhancement makes edges between bone, tissue, and fluids more distinguishable.
- **Microscopy:**
  Biological samples often have low natural contrast; scalar multiplication is used to make cells and microorganisms clearly visible.

Automotive & Transportation

- **License Plate Recognition (LPR):**
  Cameras on toll booths and traffic lights use brightness correction to handle shadowed plates, and contrast enhancement to read characters even in glare.
- **Driver Assistance Systems:**
  Contrast-enhanced video helps AI detect pedestrians, road lines, and signs in fog or at night.

TYPES OF MATRIX

1. Addition of Matrices (Image Pixel Enhancement)

Scenario:

You are enhancing a grayscale image by adding a brightness matrix to the original image matrix. Given the following matrices:

Original Image Matrix (3x3):

$$A = \begin{bmatrix} 120 & 130 & 140 \\ 150 & 160 & 170 \\ 180 & 190 & 200 \end{bmatrix}$$

Brightness Boost Matrix (3x3):

$$B = \begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

Compute the enhanced image matrix after addition.

2.Subtraction of Matrices (Image Noise Removal)

Scenario:

You are given a 3x3 grayscale image matrix representing the pixel intensities of an image. However, there's an unwanted background noise matrix that needs to be subtracted to restore clarity.

Given:

Original Image Matrix (I):

$$I = \begin{bmatrix} 120 & 130 & 125 \\ 140 & 150 & 135 \\ 128 & 138 & 132 \end{bmatrix}$$

Noise Matrix (N):

$$N = \begin{bmatrix} 10 & 5 & 8 \\ 7 & 5 & 6 \\ 5 & 7 & 4 \end{bmatrix}$$

Compute the subtracted image matrix (I - N) to remove noise from the image.

3. Matrix Product (Image Transformation Kernel Application)

Scenario:
You are applying a 2x2 sharpening kernel filter to a 2x2 image patch to enhance its features. This involves matrix multiplication of the kernel with the image patch.

Given:
Image Patch (P):

$$P = \begin{bmatrix} 100 & 120 \\ 110 & 130 \end{bmatrix}$$

Sharpening Kernel (K):

$$K = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Compute the product of kernel and image patch (K * P) to get the transformed patch.

4.Transpose of a Matrix (Image Rotation Preparation)

Scenario:
You are preparing a 3x2 pixel matrix of an image for a rotation operation. The first step is to transpose the matrix to switch rows with columns.

Given:
Original Matrix:

$$A = \begin{bmatrix} 100 & 120 \\ 130 & 140 \\ 150 & 160 \end{bmatrix}$$

Compute the transpose of matrix A to prepare it for rotation.

5.Determinant of Matrix (Color Channel Area Calculation)

In an AI-based computer graphics pipeline, a transformation matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

is applied to render 3D objects onto a 2D display.
Calculate the determinant det(A)det(A) and state what its value indicates about the transformation.

$$\begin{aligned} \det(A) &= 1 \cdot (5 \cdot 9 - 6 \cdot 8) - 2 \cdot (4 \cdot 9 - 6 \cdot 7) + 3 \cdot (4 \cdot 8 - 5 \cdot 7) \\ &= 1 \cdot (45 - 48) - 2 \cdot (36 - 42) + 3 \cdot (32 - 35) \\ &= 1 \cdot (-3) - 2 \cdot (-6) + 3 \cdot (-3) \\ &= -3 + 12 - 9 = 0. \end{aligned}$$

•The transformation is **singular** (not invertible).
•Geometrically it **collapses 3D space to a lower dimension** (volume scaling factor = 0), so depth/spatial information is lost.

6.Scalar Multiplication (Contrast Adjustment of Image)

Scenario:
An image segment is stored as a 3x3 pixel intensity matrix. You are required to enhance its contrast by a factor of 1.5 using scalar multiplication.

Given:
Image Segment (S):

$$S = \begin{bmatrix} 80 & 90 & 100 \\ 110 & 120 & 130 \\ 140 & 150 & 160 \end{bmatrix}$$

Compute the new matrix after scalar multiplication with 1.5 for contrast enhancement.

7.Matrix Dimension Identification (Image Data Compatibility Check)

Scenario:
You are combining two image layers for a graphics project. You are asked to verify whether their matrix dimensions match for a valid addition operation.
Given:
Layer 1 Matrix: 3x3
Layer 2 Matrix: 3x2

Check if Layer 1 and Layer 2 can be added. If not, explain why based on matrix dimension rules.

Matrix addition requires that **both matrices have the same number of rows and columns**.

- **Layer 1** has dimensions **3 × 3**
- **Layer 2** has dimensions **3 × 2**

Because **columns do not match, addition is not possible**.

8.Brightness & Contrast Manipulation with Transpose

Scenario:

A 2x3 grayscale image matrix is given. Find

1. Transpose the matrix.
2. Increase brightness by adding 20 to each pixel.
3. Enhance contrast by multiplying the resultant matrix by 1.2.

Given:

Image Matrix:

$$I = \begin{bmatrix} 100 & 110 & 120 \\ 130 & 140 & 150 \end{bmatrix}$$

Perform the sequence of operations and display the final manipulated matrix.

Karunya

ARISE AND SHINE