Matrices

30/10/2021

Matrix Notation

- A Matrix consists of a rectangular array of elements represented by a single symbol. [A] is the notation for the matrix and a_{ij} designates individual element of the matrix.

$$\begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \dots & a_{ij} & \dots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}$$

- A horizontal set of elements is called a row and a vertical set is called a column. The first subscript *i* always designates the number of the row in which the element lies. The second subscript *j* designates the column. For example, element a_{23} is in row 2 and column 3.
- The matrix has n rows and m columns and is said to have a dimension of n by m (or n x m). It is referred to as an n by m matrix.

Matrix Notation

- Matrices with row dimension n=1, such as, [B] = [b_1 b_2 ... b_m], are called row vectors.
- Matrices with column dimension m = 1, such as are referred to as column vectors

$$[C] = \begin{bmatrix} c_1 \\ c_2 \\ c_i \\ \cdot\cdot \\ c_m \end{bmatrix}$$

Special Types of Matrices

- Matrices where n = m are called square matrices. For example, a 3 by 3 matrix is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

- A *symmetric matrix* is the one where $a_{ij} = a_{ji}$ for all i's and j's.

For example $\begin{bmatrix} 5 & 1 & 2 \\ 1 & 3 & 7 \\ 2 & 7 & 8 \end{bmatrix}$

Special Types of Matrices

- A *diagonal matrix* is the a square matrix where all elements off the main diagonal are equal to zero, as in

$$\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}$$

- An *identity matrix* is a diagonal matrix where all elements on the main diagonal are equal to 1, as in

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Special Types of Matrices

- Transpose of a matrix is obtained by switching the row elements with the column elements. We denote the transpose of a matrix A by A^T

  For example,

  $$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ then } A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

- An *upper triangular matrix* is the one where all the elements below the main diagonal are zero
- A *lower triangular matrix* is the one where all elements above the main diagonal are zero

Matrix Operating Rules

- Addition of two matrices, [A] and [B], is accomplished by adding corresponding terms in each matrix. The elements of the resulting matrix [C] are computed,

  cij = $a_{ij} + b_{ij}$ for i = 1, 2, . . ., n and j – 1, 2, . . ., m.

- Similarly, the subtraction of two matrices, [A] minus [B], is obtained by subtraction corresponding terms,

  cij = $a_{ij} - b_{ij}$ for i = 1, 2, . . ., n and j – 1, 2, . . ., m.

- It follows directly from the above definition that addition and subtraction can be performed only between matrices having the same dimensions

Multiplication of Matrices

- The product AB (or $A \cdot B$) of two matrices A and B is defined only when:
  - The number of columns in A is equal to the number of rows in B.

Multiplication of Matrices

- This means that, if we write their dimensions side by side, the two inner numbers must match:

Matrices	A	B
Dimensions	$m \times \boxed{n}$	$\boxed{n} \times k$

Columns in A Rows in B

If the dimensions of A and B match in this fashion, then the product AB is a matrix of dimension $m \times k$.

Matrix Operating Rules

- The product of two matrices is represented a [C] = [A][B], where the elements of [C] are defined as

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

where n= the column dimension of [A] and the row dimension of [B].

That is, the c_{ij} element is obtained by adding the product of individual elements from the i^{th} row of the first matrix, in this case [A], by the j^{th} column of the second matrix [B]

Multiplying Matrices

If we define $C = AB = [c_{ij}]$, the entry c_{11} is the inner product of the first row of A and the first column of B:

$c_{11} = a_{11}.b_{11} + a_{12}.b_{21}$

Entry	Product	Value	Product Matrix
c_{11}	$\begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -1 & 5 & 2 \\ 0 & 4 & 7 \end{bmatrix}$	$1 \cdot (-1) + 3 \cdot 0 = -1$	

Multiplying Matrices

Entry	Product	Value	Product Matrix
c_{12}	$\begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -1 & 5 & 2 \\ 0 & 4 & 7 \end{bmatrix}$	$1 \cdot 5 + 3 \cdot 4 = 17$	$\begin{bmatrix} -1 & 17 & \\ & & \end{bmatrix}$
c_{13}	$\begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -1 & 5 & 2 \\ 0 & 4 & 7 \end{bmatrix}$	$1 \cdot 2 + 3 \cdot 7 = 23$	$\begin{bmatrix} -1 & 17 & 23 \\ & & \end{bmatrix}$

Multiplying Matrices

Entry	Product	Value	Product Matrix
c_{21}	$\begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -1 & 5 & 2 \\ 0 & 4 & 7 \end{bmatrix}$	$(-1)\cdot(-1) + 0\cdot 0 = 1$	$\begin{bmatrix} -1 & 17 & 23 \\ 1 & & \end{bmatrix}$
c_{22}	$\begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -1 & 5 & 2 \\ 0 & 4 & 7 \end{bmatrix}$	$(-1)\cdot 5 + 0\cdot 4 = -5$	$\begin{bmatrix} -1 & 17 & 23 \\ 1 & -5 & \end{bmatrix}$
c_{23}	$\begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -1 & 5 & 2 \\ 0 & 4 & 7 \end{bmatrix}$	$(-1)\cdot 2 + 0\cdot 7 = -2$	$\begin{bmatrix} -1 & 17 & 23 \\ 1 & -5 & -2 \end{bmatrix}$

Multiplying Matrices

- However, the product *BA* is not defined—because the dimensions of *B* and *A* are 2 x 3 and 2 x 2.
- The inner two numbers are not the same.
- So, the rows and columns won’t match up when we try to calculate the product.

Example

- $\begin{bmatrix} 3 & 1 \\ 8 & 6 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 5 & 9 \\ 7 & 2 \end{bmatrix}$
- $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

$$\begin{bmatrix} 22 & 29 \\ 82 & 84 \\ 28 & 8 \end{bmatrix}$$

Application of Matrices in digitizing images

Computer Graphics

- One important use of matrices is in the digital representation of images.

  - A digital camera or a scanner converts an image into a matrix by dividing the image into a rectangular array of elements called pixels.

  - Each pixel is assigned a value that represents the color, brightness, or some other feature of that location.

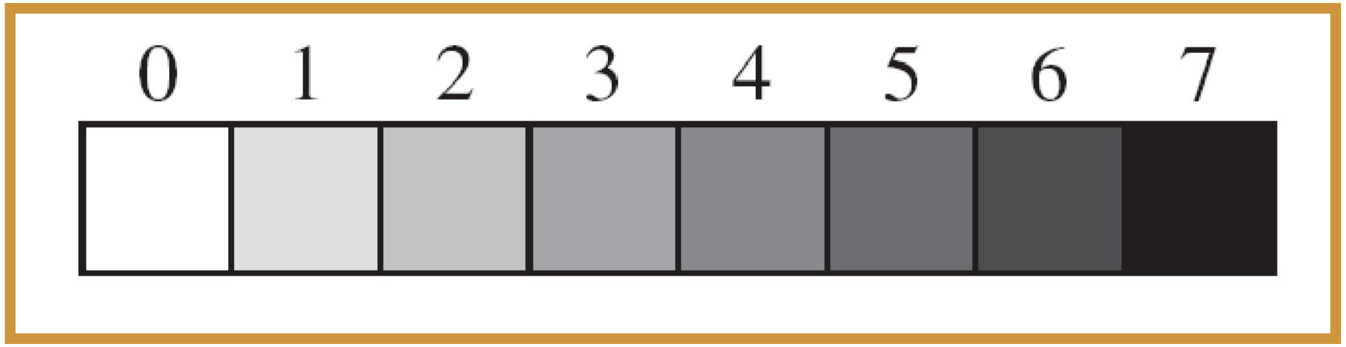
Pixels

- For example, in a 256-level gray-scale image, each pixel is assigned a value between 0 and 255.
  - 0 represents white.
  - 255 represents black.
  - The numbers in between represent increasing gradations of gray.

Gray Scale

- The gradations of a much simpler 8-level gray scale are shown.

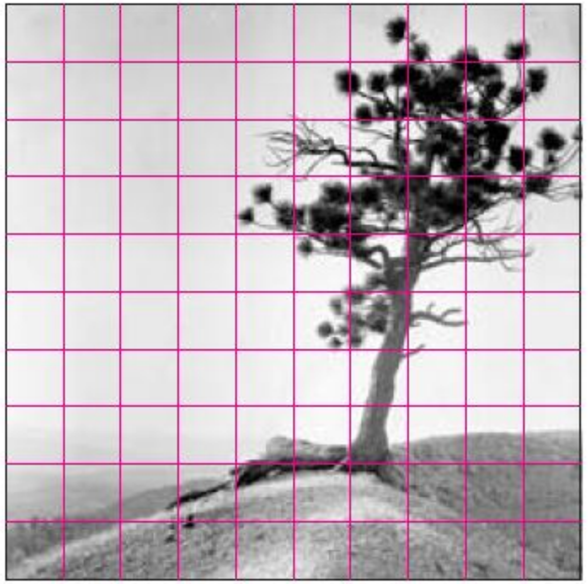
  - We use this 8-level gray scale to illustrate the process.



Digitizing Images

- To digitize the black and white image shown, we place a grid over the picture.

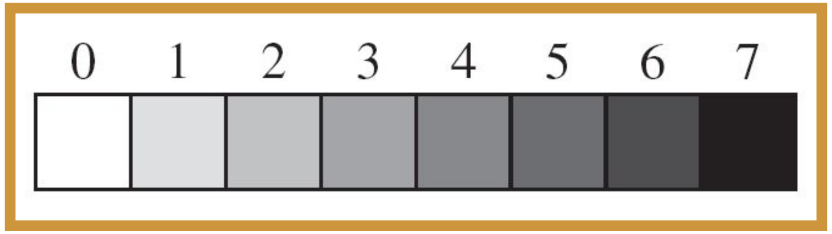
(a) Original image

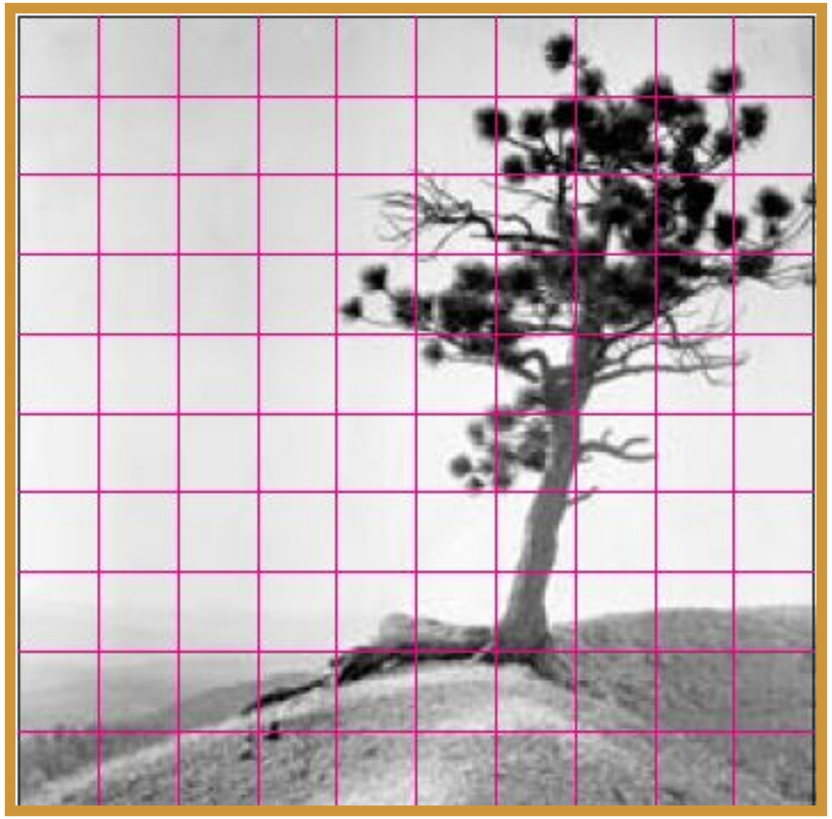


(b) 10×10 grid

Digitizing Images

- Each cell in the grid is compared to the gray scale.
  - It is assigned a value between 0 and 7, depending on which gray square in the scale most closely matches the "darkness" of the cell.
  - If the cell is not uniformly gray, an average value is assigned.



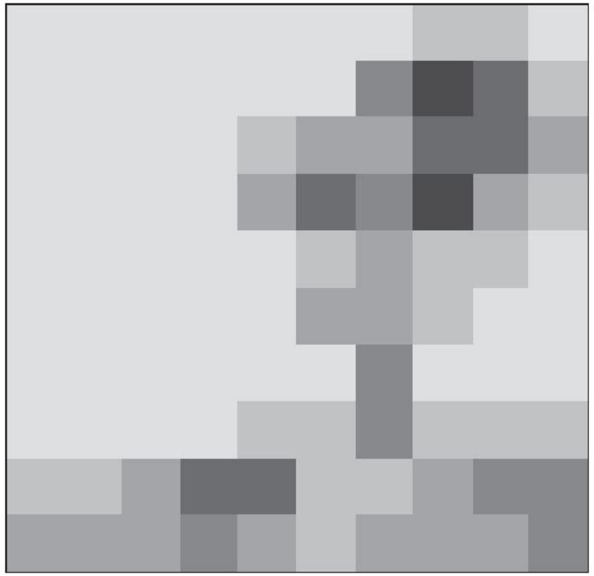


Digitizing Images

- The values are stored in the matrix shown.
  - The digital image corresponding to this matrix is shown in the accompanying image.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 6 & 5 & 2 \\ 1 & 1 & 1 & 1 & 2 & 3 & 3 & 5 & 5 & 3 \\ 1 & 1 & 1 & 1 & 3 & 5 & 4 & 6 & 3 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 3 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 2 \\ 2 & 2 & 3 & 5 & 5 & 2 & 2 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 3 & 4 \end{bmatrix}$$

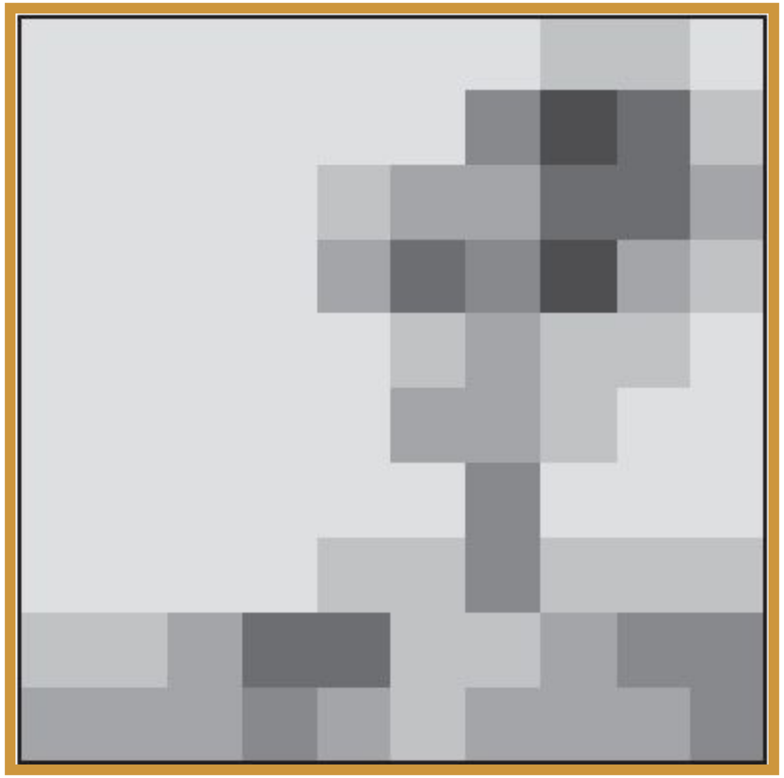
(c) Matrix representation



(d) Digital image

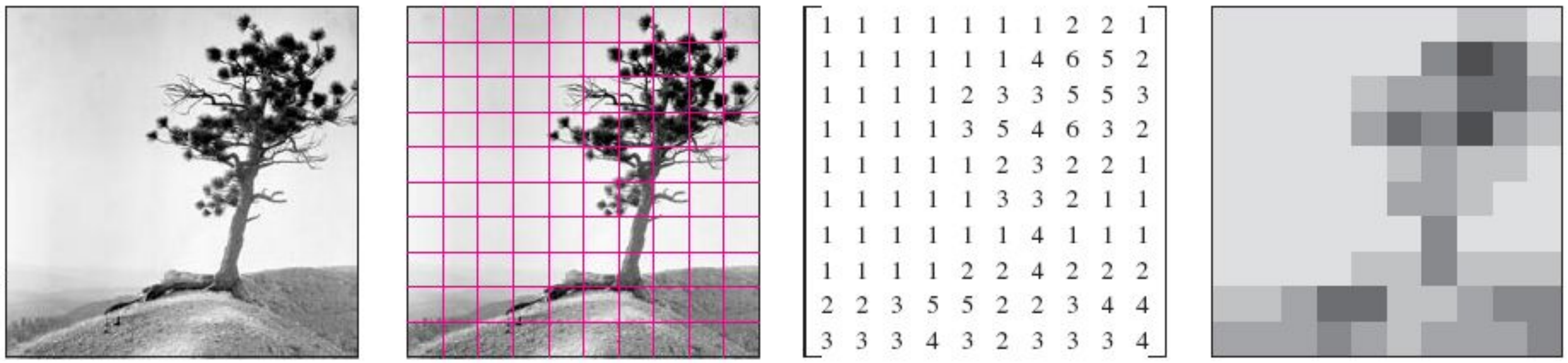
Digitizing Images

- Obviously, the grid we have used is far too coarse to provide good image resolution.
  - In practice, currently available high-resolution digital cameras use matrices with larger dimensions



Digitizing Images

- Here, we summarize the process.



(a) Original image (b) 10×10 grid (c) Matrix representation (d) Digital image

Manipulating Images

- Once the image is stored as a matrix, it can be manipulated using matrix operations.
  - To darken the image, we add a constant to each entry in the matrix.
  - To lighten the image, we subtract.

Manipulating Images

- To increase the contrast, we darken the darker areas and lighten the lighter areas.
  - So, we could add 1 to each entry that is 4, 5, or 6 and subtract 1 from each entry that is 1, 2, or 3.
  - Note that we cannot darken an entry of 7 or lighten a 0.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 6 & 5 & 2 \\ 1 & 1 & 1 & 1 & 2 & 3 & 3 & 5 & 5 & 3 \\ 1 & 1 & 1 & 1 & 3 & 5 & 4 & 6 & 3 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 3 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 2 \\ 2 & 2 & 3 & 5 & 5 & 2 & 2 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 3 & 4 \end{bmatrix}$$

Modifying Matrices

- Applying this process to the earlier matrix produces a new matrix.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 6 & 5 & 2 \\ 1 & 1 & 1 & 1 & 2 & 3 & 3 & 5 & 5 & 3 \\ 1 & 1 & 1 & 1 & 3 & 5 & 4 & 6 & 3 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 3 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 2 \\ 2 & 2 & 3 & 5 & 5 & 2 & 2 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 3 & 4 \end{bmatrix}$$

(c) Matrix representation

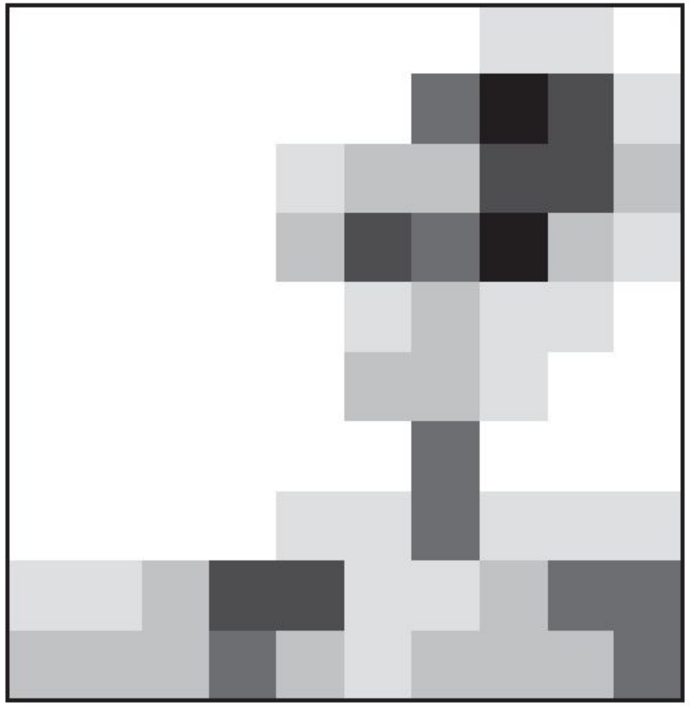
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 7 & 6 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 & 6 & 6 & 2 \\ 0 & 0 & 0 & 0 & 2 & 6 & 5 & 7 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 5 & 1 & 1 & 1 \\ 1 & 1 & 2 & 6 & 6 & 1 & 1 & 2 & 5 & 5 \\ 2 & 2 & 2 & 5 & 2 & 1 & 2 & 2 & 2 & 5 \end{bmatrix}$$

(a) Matrix modified to increase contrast

Modifying Images

- That generates the high-contrast image in the figure

(d) Digital image



(b) High-contrast image

One Hot Encoding

One Hot Encoding

- Sometimes in datasets, we encounter columns that contain categorical features (string values)
- Consider the data where fruits and their corresponding categorical value and prices are given.

Original data

Fruit	Category	Price
Apple	1	5
Mango	2	10
Apple	1	15
Orange	3	20

One hot encoded data

Apple	Mango	Orange	Price
1	0	0	5
0	1	0	10
1	0	0	15
0	0	1	20

One Hot Encoding

- Though this approach eliminates the hierarchy/order issues but does have the downside of adding more columns to the data set. It can cause the number of columns to expand greatly if you have many unique values in a category column. In the above example, it was manageable, but it will get really challenging to manage when encoding gives many columns.