Monday CVIP 04-11-24

Mapping

→ Correlation b/w adjacent pixels i.e. the values of neighbors of an observed pixel can be predicted from the value of observed pixel (interpixel redundancy)

→ Two Mapping techniques:
1. Run length coding
2. Difference coding

→ LZW Coding:
LZW assigns fixed length code words to variable length sequences of source symbols.
→ removes interpixel redundancy.
→ GIF, TIFF, PDF coding schemes used

Bit-plane Slicing:
→ removes interpixel redundancy

Binary Image Compression:
1- Constant Area Coding
2- 1-D run length Coding

Lossy Compression

→ When decompressed, the result image is different from original but close enough to original

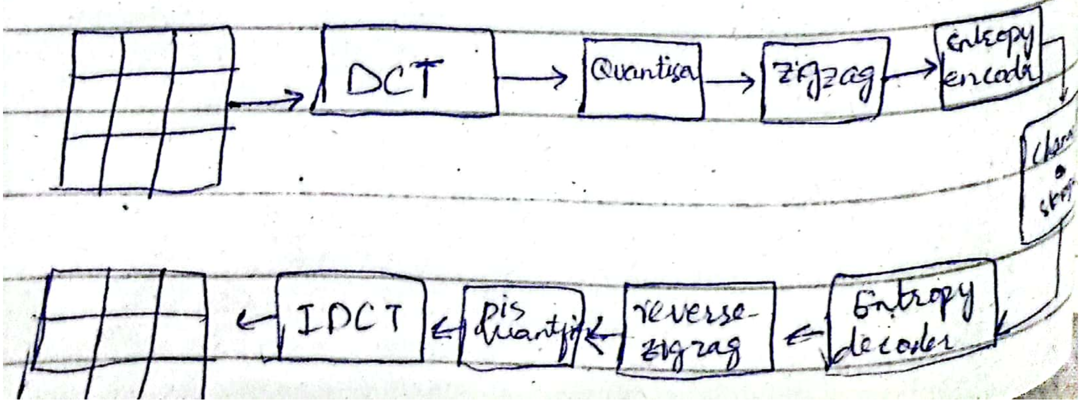
→ mostly used to compress multimedia data (audio, video, still images) especially in streaming media and internet telephony.

JPEG

→ based on use of discrete Cosine transform (DCT).

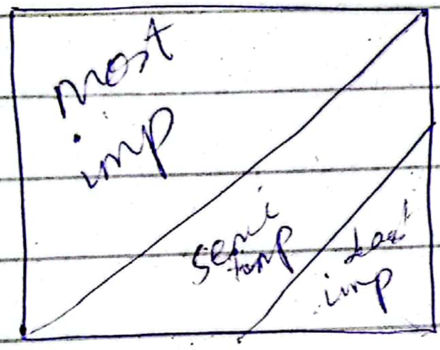
→ DCT similar to fourier transform in sense that it produces a li… of spatial frequency spectrum

JPEG Compression



→ most imp. values to our eyes will be placed in the upper left corner of matrix.

→ least imp values will be in the lower right corner.



Some Common Image Formats

JPEG → joint photographic experts group

→ JPEG is not reversible due to quantification.

LZW-coding (Lempel-Ziv-Welch)

→ LZW is loss less compression

→ TIFF → LZW

→ GIF → LZW (reversible)

Graphic interchange format

lossey

Friday :   08-11-2024

Detection of discontinuities

Three types of discontinuities:

→ Points

→ Lines

→ Edges

→ **Points detection:**

-1	-1	-1
-1	8	-1
-1	-1	-1

→ gives high weight to contral pixel. and less or nullify neighboring.

→ **Line Detection:** (Apply all mask ⇒ H Task)

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal

-1	-1	2
-1	2	-1
2	-1	-1

+45°

-1	2	-1
-1	2	-1
-1	2	-1

Vertical

2	-1	-1
-1	2	-1
-1	-1	2

-45°

Morphology and Morphological I[P]

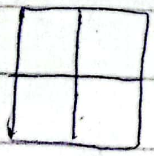
(moving objects) Foreground : White color (region of i[nterest])
Background : Black color (non-moving objects)

Morph : forms or shapes
ology : to study something
Image Morphology : brach deals with fo[rm] and structure of images.
→ used to extract the image components for representations

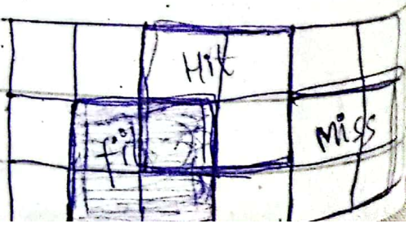
Dilation and Erosion

→ Morpholog deals with structuring el[ement] (SE)
→ Fit : All pixels in SE cover on pixels in the image.
→ Hit : Any pixel in SE covers pixels in the image.
→ Miss : All are missed

Structuring element ⇒ same as spatial filters (may have any shapes & s[izes])



S.E



→ SE can have v[alues] co-efficients.
→ For simplicity we [use] structuring elements at the middle pi[xel]

1	1	1
1	1	1
1	1	1

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

→ Morphological I[P] filtering.
→ SE is moved in the original pixel in a n[ew]
→ value of this on the operat[ion]

Erode : If SE
Dilate : If SE

Erosion =

SE can have varied values of co-efficients.
For simplicity we will use rectangular
ucturing elements with their origin
the middle pixel.

1	1	1
1	1	1
	1	1

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

→ Morphological IP is same as spatial filtering.

→ SE is moved across every pixel in the original image to give a pixel in a new processed image.

→ value of this new pixel depends on the operation performed.

SE → structuring element

Erode : If SE Fits then it is Erode

Dilate : If SE HIT then it is dilate

Erosion = $t \ominus S$ $\quad$ $t \rightarrow$ image, $S \rightarrow$ S.E

Erosion:

$$g(x,y) = \begin{cases} 1 & \text{if } S \text{ fits } f \\ 0 & \text{otherwise.} \end{cases}$$

→ new pixel is determined by above rule.

→ Erosion can split apart joined parts

→ Erosion can strip away extrusion

→ Erosion shrinks objects.

Dilation:

image

S.E

$$t \oplus S$$

$$g(x,y) = \begin{cases} 1 & \text{if } S \text{ hits } f \\ 0 & \text{otherwise} \end{cases}$$

image

hits mein fit ho ga already

→ 1 if hit or fit in case of dilation

→ Broken characters are joined.

Applications

Dilations

→ can repair breaks

→ can repair intrusions

→ enlarge objects

Compound Operations

→ mostly morphological operations can be performed by combining erosions and dilation.

Mostly

i- Opening

ii- Closing

→ Opening:

$$t \circ s = (t \ominus s) \oplus s$$

→ Note a disc shape SE is used.

→ Closing:

$$t \cdot s = (t \oplus s) \ominus s$$

Monday 11-11-24

Connected Component Extraction:

1 → Find first pixel of object

2 - Apply 3X3 window.

3 - Apply all pixels having intensity 1 and push in stack. (clockwise or anticlock

4 - Change color of pixel at which window is applied.

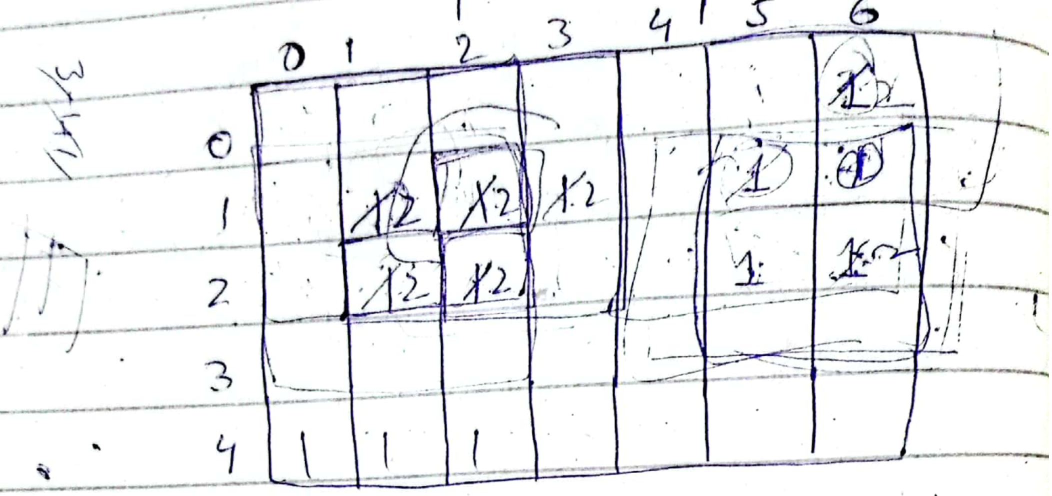
5 - Put that pixel on point array.

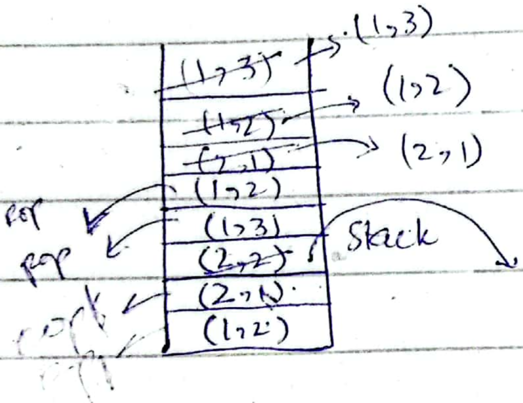
6 if (!stack.empty) {

7 Pop from stack = P

if (P color == 1)

repeat step 2-6



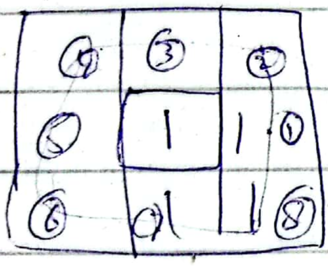


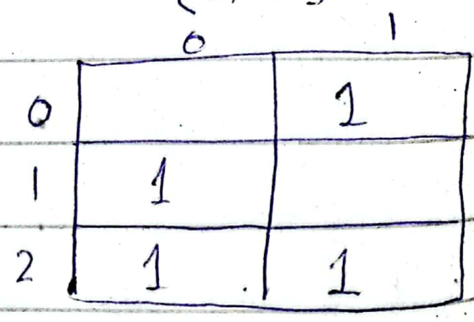
point array

points $\{(1,1), (2,2), (2,1), (1,2), (1,3)\}$

P(2,2)

P_1 $\{(0,6), (1,5), (2,6), (2,5)\}$





(x-minx, y-miny) for new coordinates ←

(0,6) → (0,1)

(1,5) → (1,0)

(2,5) → (2,0)

(2,6) → (2,1)

3D

Min X = 0 , Max X = 2

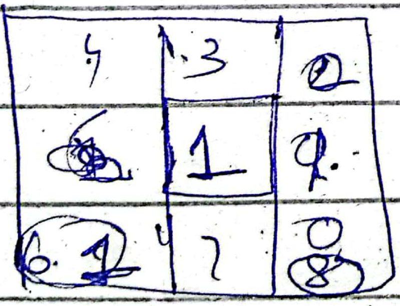
H = X = max X − Min X + 1

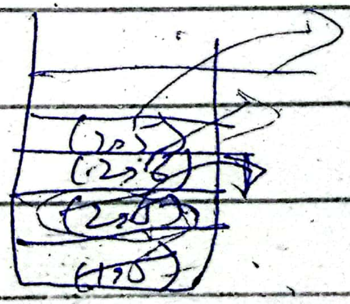
= 2 − 0 + 1 ⇒ 3 → No of rows

Min Y = 5 , Max ~~X~~ Y = 6

W = Y = max Y − min Y + 1

= 6 − 5 + 1 ⇒ 2 → No of columns



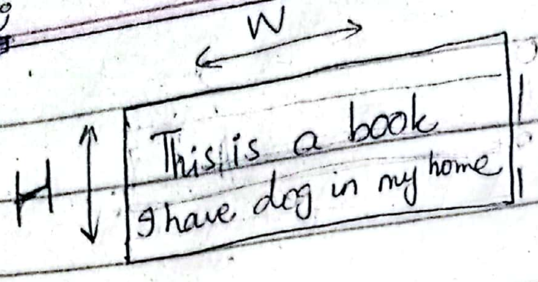


(2,6)

Friday

CVIP

22-11-24



CC

T

II

I

S

Lines

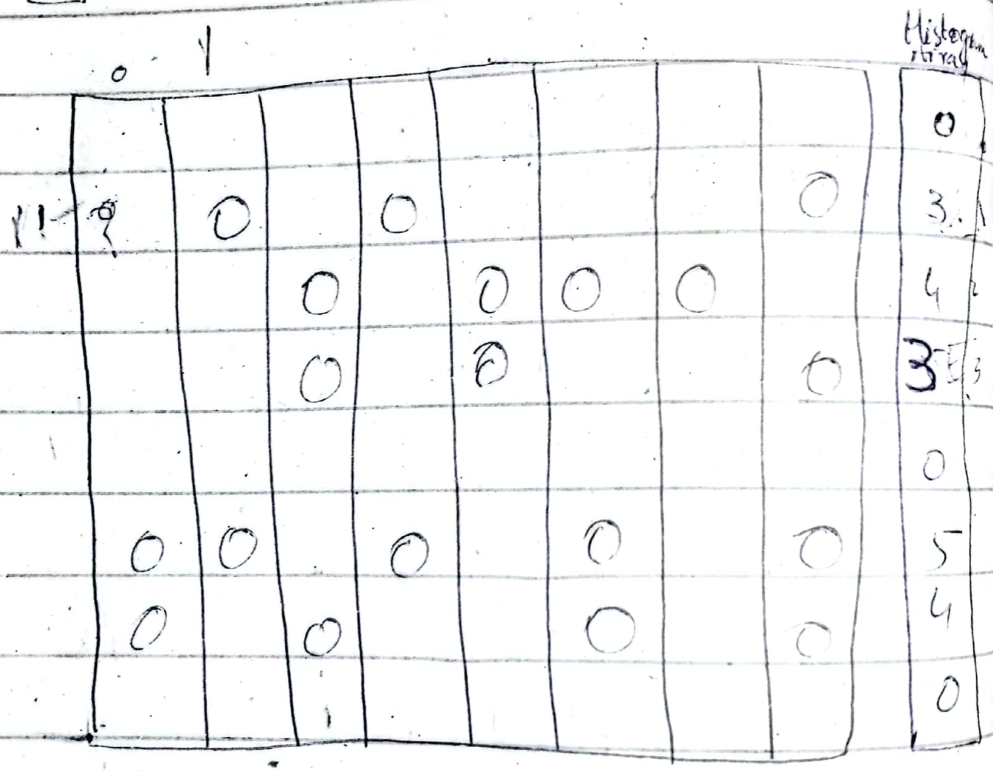
This is a book.

I have dog in my home

$h = e - s + 1$

$w = W$

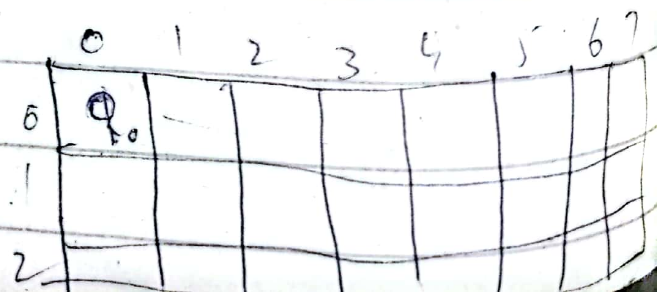
words



$W = 8$

$H = 8$

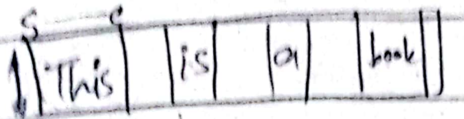
$h = 3 - 1 + 1 = 3$



for line sol

```
for (r = s to e)
    for (c = 0 to W)
        Line [r - s][c] = I [r][c]
```

for words!



$h_1 = h$

$W_1 = e - s + 1$

```
for (r = 0 to h1)
    for (c = s to ~~h W1~~ e)
```

Convolution Neural Network

→ Used in areas:

1. image recognition
2. image classification
3. object detection
4. Face recognition etc

→ CNN takes

- an input images
- process it
- classify it under certain category

→ Computers see image as array of pixels:

image resolution = $h \times W \times d$

d ↑ dimension

RGB (3)

Series of Convolution layers:
→ filters (kernels)
→ pooling
→ Fully Connected layers (FC)
→ apply softmax function to classify an object with prob. from 0 to 1.

Input → Convolution + ReLU → Pooling → Convolution + ReLU → Pooling

feature learing

→ Flatten → Fully Connected → Softmax → output

2D to 1D

Classification

Convolution Layer

→ first layer to extract features of input image.
→ takes two input such as image matrix and kernel.
→ preserves relationship b/w pixels by learning image features using small square of input data.

outputs a volume dimension $= (h - f_h + 1) \times (W - f_w + 1)$

$$\begin{array}{c|cccccc|}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} * \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$1 = (1)(1) + (1)(0) + (0)(0) + (1)(1)$

$= 1 + 0 + 0 + 1 = 2$

$2 = (1)(0) + (1)(1) + (1)(0) + (0)(1) + (1)(0) + (1)(1$

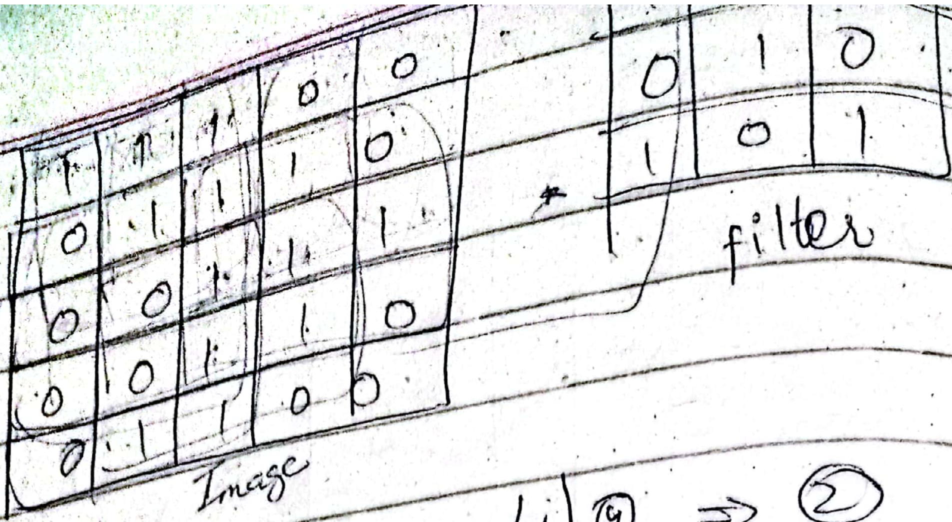
$= 0 + 1 + 0 + 0 + 0 + 1 = 2$

$3 = 1 + 1 + 1 = 3$

$4. = 1$

$5 = 1\ \cancel{4}$

$6 = 1$



① ⇒ 1 + 1 + 1 + 1 = 4

(1+1+1)

② ⇒ 3

(1+0+0+0+1+0+0+0+1)

③ ⇒ 4

④ ⇒ 2

⑤ ⇒ 4

⑥ ⇒ 3

⑦ ⇒ 2

⑧ ⇒ 3

⑨ ⇒ 4

5 - 3 + 1 =

4	3	4
2	4	3
2	3	4

feature Map

Types of filters

→ Edge detection

→ Blur

→ sharp

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

Edge detection $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Strides:
No. of pixels shifts over the input matrix (jump type something

→ When 2 strides then move filter to 2 pixels.

*→ for 7X7 image and 3X3 filter

Strides	output image dimens
1	5X5
2	3X3

output size = $\boxed{(N-F)/\text{stride} + 1}$

N = image ⇒ 7
F = filter ⇒ 3

Stride 1 ⇒ $(7-3)/1 + 1 = 4+1 \Rightarrow 5$
Stride 2 ⇒ $(7-3)/2 + 1 = 4/2+1 \Rightarrow 3$
Stride 3 ⇒ $(7-3)/3 + 1 = 4/3 + 1 \Rightarrow 7/3 = 2.33$
Stride 4 ⇒ $(7-3)/4 + 1 = 1+1 \Rightarrow 2$

Filter does no...

Padding: pad the picture with zeros
→ pad
→ Drop that part of the image where the filter did not fit
This is called valid padding as it keeps only valid part of im...

0 – 2

Friday:s

6 6

24+3-1+

0 2 -4

12+3̶9̶4+12 92

~~24+6-8+12~~

92

Lin[0]

82

horizontal

3	10	5
0	0	0
-3	10	3

92

4

9
12
24
12
32

$n-f+1 \times n-f+1$

$\boxed{n+2p-f+1} \times n+2p-f+1$

6

filter does not fits

Padding:

→ pad the picture with zero

→ Drop that part of the image where the filter did not fit

This is called valid padding as it keeps only valid part of

Friday:s

6 6

0 2 -4

24+3-1+

~~24+6-8+12~~ 12+3

92

Lin[0][0] 48

72

82

horizontal

3	10	3
0	0	0
-3	10	-3

92

9

12

24

12

32

$n-f+1 \times n-f+1$

use floor for point value

02x12 = 24

Monday:

CVIP

→ Major feature extraction
→ feature map size $N = n - f + 1$
→ ReLU maps negative value to zero
→ No padding in CNN
→ For padding
filter map size = $n + 2P - f + 1 = n$

$P = \frac{f-1}{2}$

R

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

7x7

*

3	4	4
1	0	2
-1	0	3

Stride by 2 & 3
convolve

For stride:

$$N = \left[\frac{n+2P-f}{S} + 1 \right] \times \left[\frac{n+2P-f}{S} + 1 \right]$$

floor the values if it's in points

→ For RGB the dimensions of image and filter should be same.

$n \times n \times d$ * $f \times f \times d$ → resultant value image

$6 \times 6 \times 3$ $\quad 3 \times 3 \times 3$

For one time Convolution = 3×9 times SOP.

$$V_{filter} = V_R + V_g + V_B$$

Assignment:

For G - first row zero

For B = Forth row zero.

for filter repeat the same filter

$n \times n \times n_c$ * $f \times f \times n_c$ → $n-f+1 \times n-f+1 \times$ no. of filters

$4 \times 4 \times 2$

Image pass krna he → page segmentation module ko

Yolo

HBL A/C (.png) → OCR → HBL A/C (.txt) → DBMS

Tesseract OCR

class ID	Training Samples	Testing Samples

ClassID	ACC

Accuracy matric

area wise accuracy matrice?

Friday: 06-12-24

CNN

Pooling → imp feature ki dimension reduce krty

Flatten → 2D to 1D.

$x w_i + b_i$

Exact

ReLU → Rectified Linear Unit for a non-linear operation.

$F(x) = \max(0, X)$

Pooling layer: → retains imp feature

reduce the no. of parameters when the images are too large.

Spatial pooling also called

→ Subsampling

→ downsampling

different types:

→ Max Pooling

→ Average Pooling

→ Sum Pooling

Stride by 2 filter 2x2

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

Min pool

1	2
1	0

Average pool

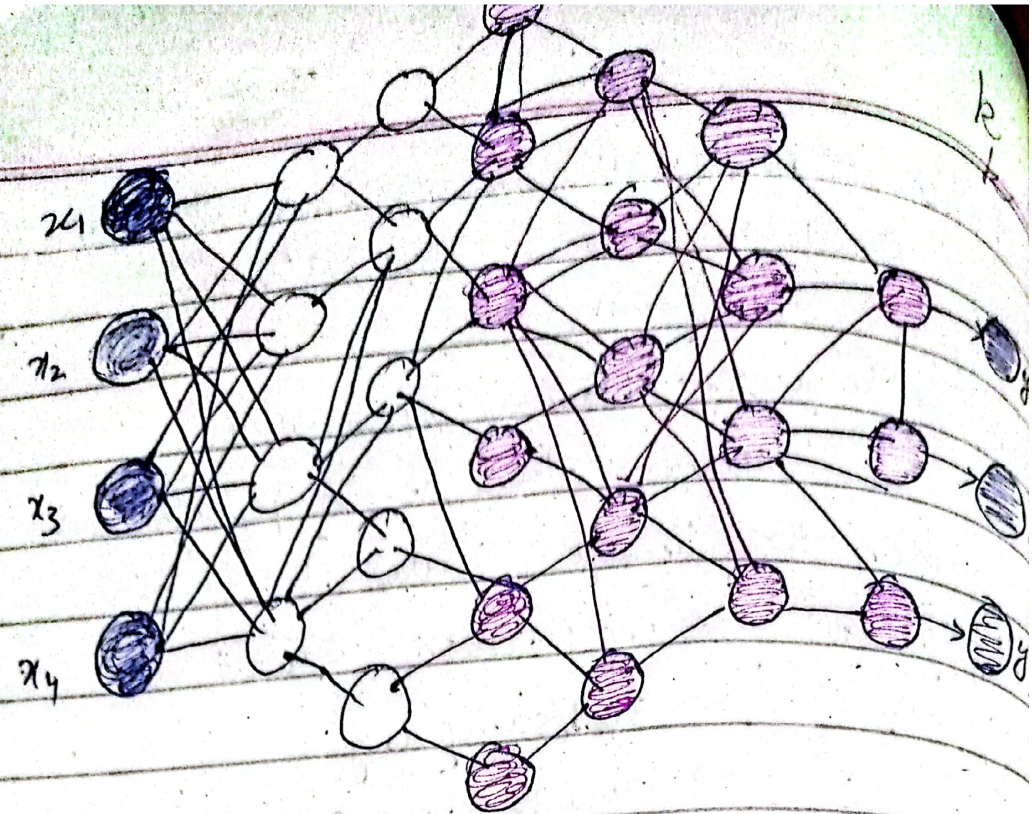
3.25 = 3	5.25 = 5
2	2

Sum pool

13	21
8	8

Max po

6	8
3	4



Softmax:

→ it turns a vector of K real values into a vector of K real values that sum to 1.

Fully Connected layer:

→ neuron applies a linear transformation to the input vector through a weights matrix.

$$y_{jk}(x) = f\left(\sum_{i=1}^{n_H} \underbrace{w_{jk}}_{\text{weight}} x_i + \underbrace{w_{j0}}_{\text{bias value}}\right)$$

non-linear transformation is then applied to the product. through a non-linear activation function.

Weight matrix

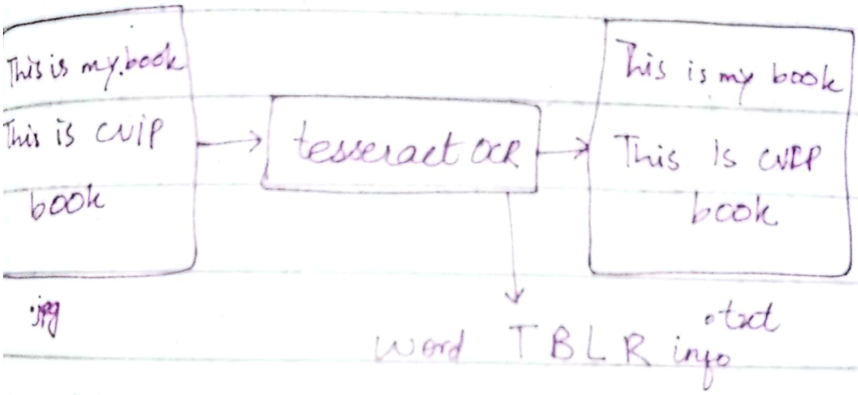
rows = No. of components in vector

Colums = No. of Classes

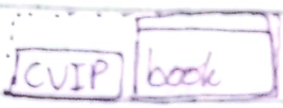
input vector · Weights matrix = output

1X9 9X4 1X4

Yolo, tesseract OCR



tomatic annotations



top = min(topi)

bottom = max(bottomi)

left = min(lefti)

right = max(righti)

for yolo

class-ID ⇒ book-Name
Label ⇒ CVIP book
TBLR ⇒ T, B, L, R

Her class k correspond color

RNN

Recurrent Neural Network:
→ Sequence learning

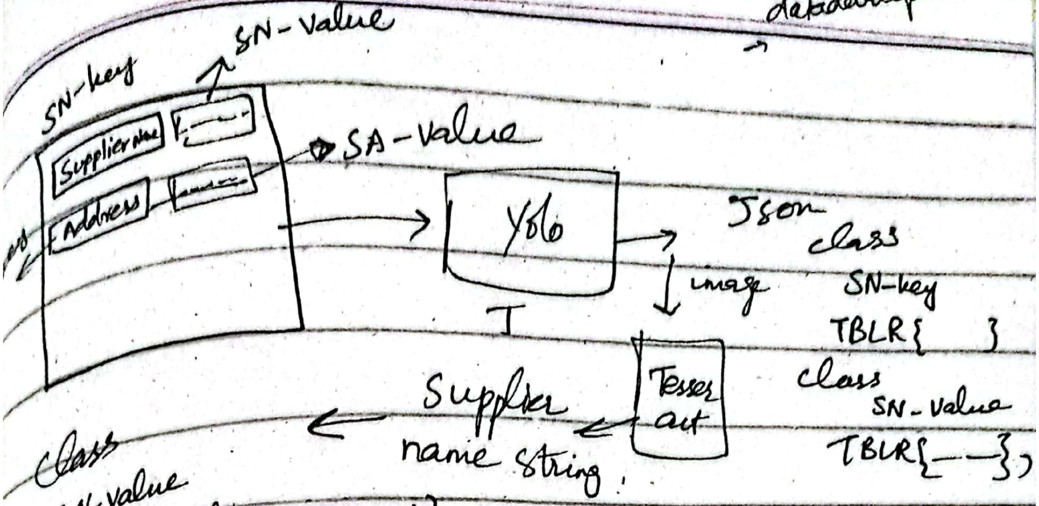
These books → are → high prob.
arc
erc
ere

Feed forward network:
→ Output is passed forward, it does not comes backward.
→ output at one layer does not affect the output on other layer.

logo embed krna.

4-format 20000 → Total entries
↓
5000 aik format hi

→ pipeline
datadevelopment
→



Class
SN-Value
TBL {volo sy aaye gi}
OCR {SSKhan}
→ Info M.S + DB

360 DB

Super Annotate :
Yolo → koko format mein input data

positional
Info for supplier name and brand name

— 18 Dec 11:59 PM
Readmefile .md.

data { Images .jpg
excel
items.

Yolo + OCR { 20

Complete { 25 - 26