CVIP - 15 (04/11/24)

Lossy Compression

1) LZW ↳ interpixel ↳ Gif / Tiff / PDF
2) Bit Plane Code.

↳ drastic size reduction. ↳ no noise

↳ freq domain compression.

Discrete Cosine Transformation (DCT)

DCT → Quantizer → zigzag → Entropy Enc

Channel/Storage ↵

IDCT ← dequantizer ← reverse zigzag ← Entropy Dec ↵

- Gif / Tiff → LZW
- JPEG → DCT

CVIP - 16 (08/11/24)

Segmentation & edges:

↳ Detection of discontinuities.

↳ point detection mask

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

↳ Line detection mask.

$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$ & Horizontal

$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$ vertical

$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$ $+45^\circ$

$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$ -45°

1) Apply filter.

2) threshold for clear results.

Task:

↳ Sample images of ___ book.

apply horizontal, vertical, 45°, -45° masks.

↳ disconnectivity 

↳ 1 pixel disconnectivity may not be inherent disconnectivity.

Morphological Image Processing:

real life: background → white, foreground → black

morphological: bg → black, foreground → white.

↳ Morph. forms or shapes

↳ Ology: study of something.

↳ Morphology: branch of biology that deal

↳ Morphological image processing works on binary image.

(strict) **fit:** Pixels in SE fit on pixel in an image.

(linear) **hit:** Any pixels in SE covers pixel in an image.

miss: All are missed.

Dilation & Erosion SE

↳ Any size or shape and any value of co-efficient.

↳ usually odd size for center pixel value.

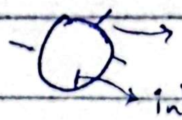
Erode: If SE fits

Dilate: If SE hits. (expands).

Erosion: $t \ominus s$

$$g(x,y) = \begin{cases} 1 & \text{if } s \text{ fits } t \\ 0 & \text{otherwise. (hit+miss)} \end{cases}$$

Input image → negative → morphological op → negative



↳ Erosion can strip away extrusions.

Dilation: $t \oplus s$

$$g(x,y) = \begin{cases} 1 & \text{if } s \text{ hits } t \\ 0 & \text{otherwise (miss only)} \end{cases}$$

When dilation? when there is thin & thick transition

↳ eg doc image.

↳ thin lines may have disconnectivity, dilation can repair it.

↳ repair intrusions.

↳ enlarge the objects.

Opening:

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

Closing:

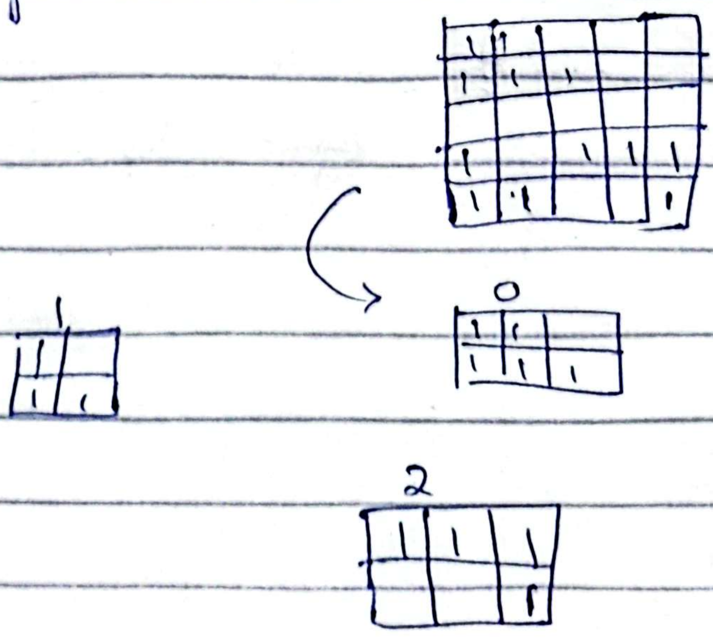
$$t \bullet s = (t \oplus s) \ominus s$$

task:

foreground = 1 bg = 0

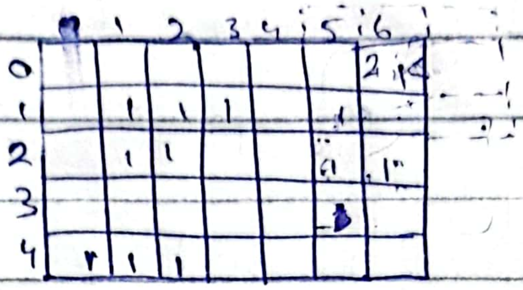
↳ object detection

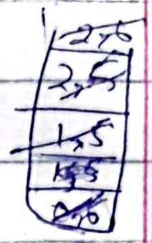
↳ edge detection.



CVIP-17 (11/11/24)

Connected Component Extraction:





↳ Apply padding to boundary pixel.

1) find first pixel of object.

2) Apply 3x3 window.

3) find all pixels having intensity and position stack. in clockwise or anticlockwise direction.

4) Change color ~~g~~ of pixel on which window is applied.

5) put that pixel in point array.

6) if (stack NotEmpty)

7) p = pop from stack

{ if (p color == 1)

Repeat step 2-6.

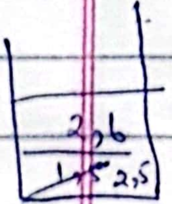
}

Struct Point{
 int x;
 int y;
}

Points = {(0,6)

Raster Scan

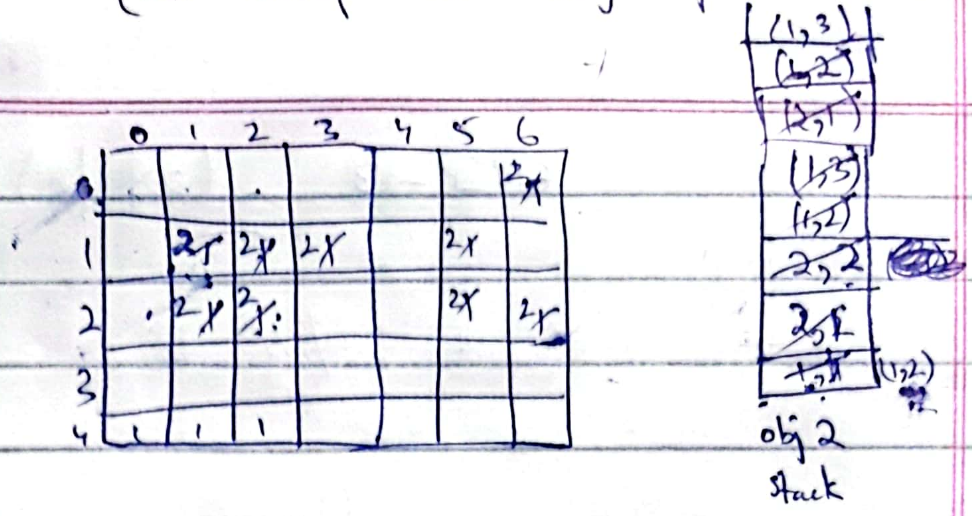
↳ Use points array to form sub image



Obj 1 stack

Obj 1 Points = {(0,6),(1,5),(2,6),(2,5)}

(color change after adding neighbours to stack)



Obj2 Points = {(1,1), (2,2), (2,1), (1,2), (1,3)}

Obj 1

	0	1
0		1
1	1	
2	1	1

(0,6) → (0,1)
(1,5) → (1,0)
(2,6) → (2,0)
(2,5) → (2,1)

Min X = 0
Min Y = 5

Max X = 2
Max Y = 6

Size of sub image:

W ~~height~~ = Max Y − Min Y + 1
= 6 − 5 + 1
= 2.

H ~~width~~ = Max X − Min X + 1
= 2 − 0 + 1.

$\boxed{\text{Add 1 cause of 0 indexing}}$

(0,6) − (Min X, Min Y) → (0,1)

↳ Extract chars from image.

CVIP -18 (22/11/24)

Assignment:

↳ use hand written text / SS to extract cc, lines & words in 3 separate folders.

Histogram (Array)

This is a book
I have dog in my house

w
h [ ] line.jpg

~~h = end - start + 1~~

for (r = s to e)
    for (c = 0 to w)
        line [r-s][c] = I [r][c]

⇒ col histogram for separating words.

CNN

Grayscale → 1D, RGB → 3D.

CNN models → Resnet, inception net, image net, mobile net etc.

→ CNN also works on text (one hot encoding).
(convolution + pooling) → pair.

→ highest prob is on top of output layer
↳ 'cat' in top 3 also used for acc.

Input → conv+relu → pooling → conv+relu → pooling
softmax ← fc layer ← flatten

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

feature Map: $\begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$

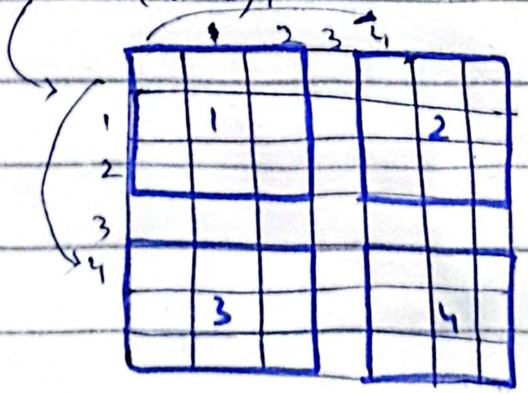
Stride:

↳ sliding window (n-pixels moved).

feature Map Dimension: $(N-F)/\text{stride} + 1$

$(7-3)/3 + 1 = 2.33$ (not possible)

$(7-3)/4 + 1 = 2$



→ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ (2×2)

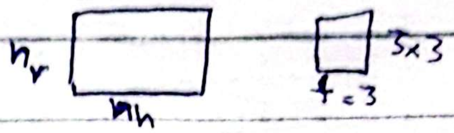
Padding:

↳ Artificial padding ⇒ synthetic info.

↳ CNN has valid padding.

CVIP- 19+20 (02/12/24)

→ feature extraction is most imp step in computer vision apps.



→ positive → white
0 → gray
negative → black.

⇒ In CNN/dependency we stack filters.

↳ separate features map for vertical lines.

↳ sep feature map for horizontal lines.

Padding:

↳ usually no padding in CNN.

↳ padding adds synthetic (false) info.

→ new image dimensions.

$n-f+1$ (without padding)

$n+2p-f+1$ (with padding)

(p= padding $p=1$ ⇒ 1row/col start+end)

→ Our intention is to get similar dimensions after filtering.

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

Stride Convolution:

$$\text{new size} = \left[\frac{n+2p-f}{s}+1\right] * \left[\frac{n-2p-f}{s}+1\right]$$

Conv on RGB

n_c of img = n_c of filter
↳ no of channels.

↳ 1 time conv of B&W = 9 SOPs.
= 9+9+9
↳ 1 time conv of RGB. = 27 SOPs.

n_c = no of channels
n_c' = no of filters (stacked)

Home Task

Conv on RGB.

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

G.

0	0	0	0	0	0	0
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

B → Row ~~3~~ (4th) is 0.

Filter

3	4	4
1	0	2
-1	0	3

× 3 channels.

Home Task:

Convolution on RGB. $\boxed{\text{Stride}: 2}$

filter:

3	4	4
1	0	2
-1	0	3

* 3 Channels.

R * filter:

G → Row 1 ⇒ 0's
B → Row 4 ⇒ 0's

(7x7)

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

*

3	4	4
1	0	2
-1	0	3

(3x3)

$$\text{Size} = \frac{7-3}{2} + 1 = 2+1 = 3\times 3$$

91				

91	100	88
69	91	117
44	72	74

1) $6+12+28+6+18-3+24 = 91$
2) $21+16+24+9+14-8+24 = 100$.
3) $18+8+36+7+6-8+21 = 88$.
4) $9+16+32+7+6-4+3 = 69$
5) $24+12+32+3+12-1+9 = 91$
6) $24+36+28+6+8-3+18 = 117$
7) $12+8+4+3+8+9 = 44$.
8) $3+32+12+4+18-3+6 = 72$.
9) $9+16+24+9+6-2+12 = 74$

G* filter

0	0	0	0	0	0	0
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

*

3	4	4
1	0	2
-1	0	3

45	39	26
69	91	117
44	72	74.

1) $6+18-3+24 = 45$
2) $9+14-8+24 = 39$
3) $7+6-8+21 = 26$

B* filter:

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

*

3	4	4
1	0	2
-1	0	3

91	100	88
56	76	103
44	72	74

1) $9+16+\cancel{2}32-4+3 = 56$
2) $24+12+32-1+9 = 76$
3) $24+36+28-3+18 = 103$

Conv Ans: (sum all)

227	239	202
194	258	337
132	216	222

Stride 3:

$$\text{New Size} = \frac{7-3}{3} + 1$$

$$= 4/3 + 1 = 1+1$$

$$= 2 \times 2.$$

R* filter:

91	84
80	93

G* filter:

45	40
80	93

B* filter:

91	84
15	39

Final Grid: (sum up).

227	208
175	225

Calculations

⇒ R* filter

1) 6+12+28+6+18-3+24 = 91
2) 12+24+8+8+8-3+27 = 84
~~24+28+16+3+18-6+9 = 92~~
3) 21+32+12+4+2-3+12 = 80
4) 18+24+12+8+8-1+24 = 93

⇒ G* filter

1) 6+18-3+24 = 45
2) 8+8-3+27 = 40
3) 21+32+12+4+2-3+12 = 80
4) 18+24+12+8+8-1+24 = 93.

⇒ B* filter

1) 6+12+28+6+18-3+24 = 91
2) 12+24+8+8+8-3+27 = 84.
~~24+28+16+3+18-6+9 = 92.~~
3) 4+2-3+12 = 15
4) 8+8-1+24. = 39.

CVIP-21 (6/12/24)

Project:

Script / labeling.

↳ Image to page segmentation

↳ How many fees of 4th semester a.

↳ Generate Challan

TBLR → Dimension of (text).

↳ Tesseract OCR.

EC2 / YOLO →

1) Auto labeling for TBLR.

2)

Page segmentation for required fields only.

Lecture:

Conv + relu → no negative value in colors → pooling.

Exact → no dimension change

Same → padding added.

→ Map Prob to 0 - 1.

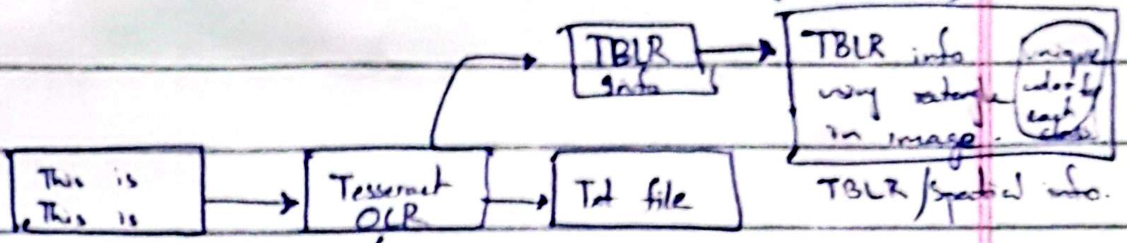
↳ Multiple filters → multiple feature maps

$\boxed{1 \times 9}$ $\boxed{9 \times 4}$

1x4 ⇒ Output vector.

→ cat & dog class using CNN.

CVIP-22 (09/12/24)



Eng txt => even formatted image

1) Layout training:

↳ Annotated data (tblr info, data)

cvip book (Auto annotate together)

$T = Min(Top_i)$

$B = Max(Bottom_i)$

$L = Min(Left)$

$R = Max(Right)$

YoLo = ClassID + TBLR (XML)

1) class_ID : book_name

2) TBLR : {T, B, L, R}

3) label : cvip book.

=> YOLO config.

RNN:

recognizer, identifies separate words and then form sentences based on previous sentences based on previous sentences.

These books are
are-ed } checked from language model.

⇒ feed forward.

CVIP - 23 (13/12/24).

Active Text Pair List (ATL).

1) row → supplier (serial no → receipt no)

2) row → buyer

3) 4 cols/rows of ~~address~~ products

4) ⇒ embed logo to pdf.

4 formats ⇒ 5k each.

incremental day invoice.

YOLO takes TBLR info.

↳ read json TBLR 1-by-1:

pdf to image

store data in tabular form.

No/Num Date:

json to coco.

super Annotator.

TBLR info by using connected comp

1) Pipeline 2) Data.

Data:

Pdf : 20k files

Jpg : 20k files

Excel file : Date 2 → Date

YOLO + OCR

Complete Syst

CVIP-24 (17/12/24)

CNN → for classification without timestamp.

RNN → classification at time T considering prediction at T-1

Feed Forward:

Output of 1 layer ⇒ each neuron of next layer.

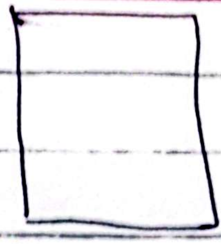
↳ 1st iteration's output is 2nd iteration's input.

1) 1-to-1 RNN

(one input image ⇒ one output).

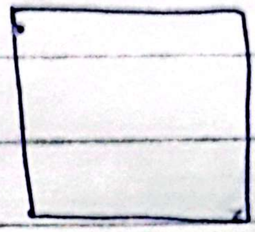
Dog image ⇒ Dog label.

2) 1-to-Many RNN.

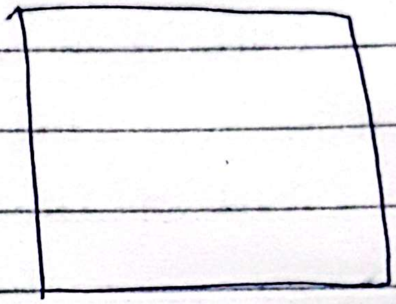


3) Many-to-1 RNN

↳ Sentiment Analysis

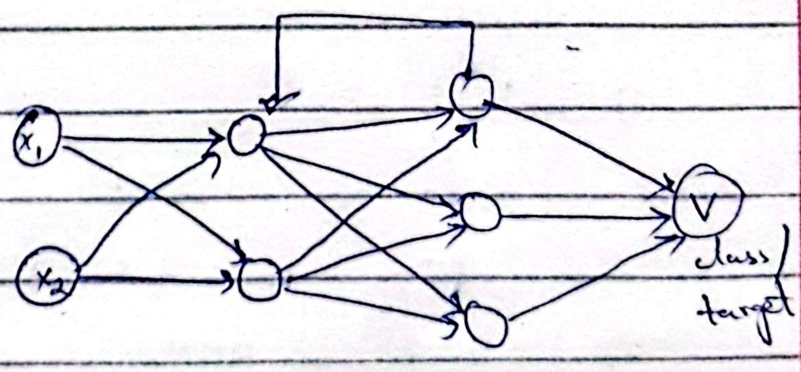


4) Many-to-Many RNN.



⇒ Sometime error is very low ⇒ when weights are re-adjusted it starts more ~~fa~~ false predictions.

↳ gradient descent etc used.



deep if > 1

Image Classification:

1) Multi Class 2) Multi label.

Image Localization:

↳ classify + location.

Image Segmentation:

↳ boundary instead of rectangle

Resnet:

Why transfer learning?

N ⇒ no of classes.

↳ lots of time & effort for training from scratch.

↳ mobile net, image net, resnet.

↳ feature extraction part requires effort ⇒ use this part & refine in fc & softmax.

CIFAR.

R. Paper ⇒ Deep Residual learning for Image Recognition.

Code.

```
def feature_extracter(inputs):
    feature_extracter_layer = tf.keras.applications.resnet.ResNet50(
                input_shape = (224, 224, 3),
                include_top = false,
```

(last layer per [illegible])

```
                weights = "imagenet") (inputs)
    return feature_extraction_layer.

def classifier(inputs):
    x = tf.keras.layers.GlobalAveragePooling2D()(inputs).
    x = tf.keras.layers.flatten()(x)
    x = tf.keras.layers.Dense(1024, activation="relu")(x)
    x = ———————— 512,
    x = ———————— 10, activation="softmax",
                        name="classification")(x).
    return x.

def final_model(inputs):
```