CS512 - COMPUTER VISION

ASSIGNMENT - 4

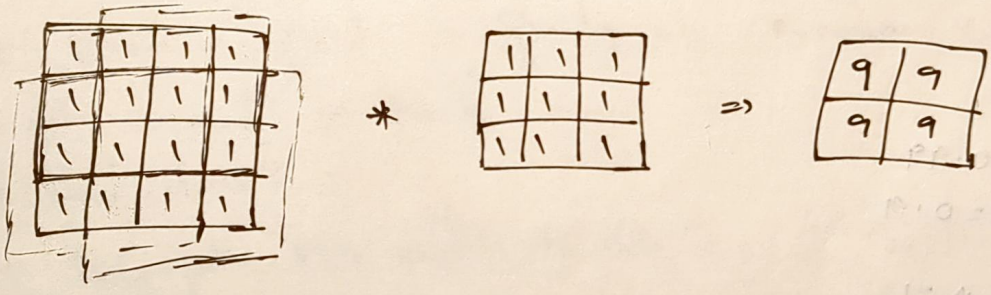
SAI MANOHAR VEMURI
A20514848

(1) Convolution Layers

a) 4X4 RGB Image

Channel R

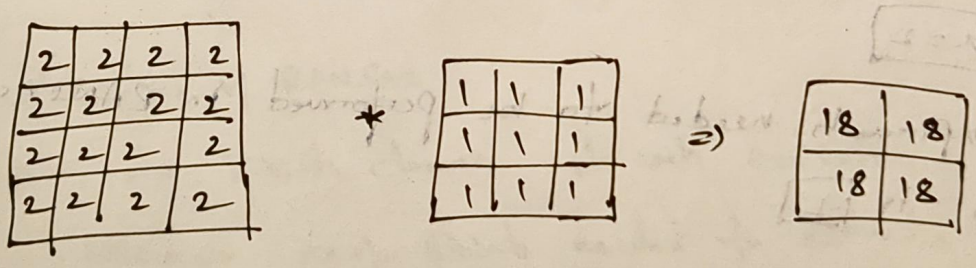
3X3 filter



first Element Conv = (1X1+1X1+1X1) * 3
= 9

Similarly, we need to move the filter by 1 position as the stride is 1. Since, all the values are same we get the same output i.e "9".

Channel G



= (2X1 + 2X1 + 2X1) * 3
= 18

Similarly, We need to apply the filter for the rest of the image pixels with the stride 1.

Channel B

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

$*$

1	1	1
1	1	1
1	1	1

First Element $= \begin{pmatrix} 1\times1 + 1\times1 + 1\times1 + \\ 2\times1 + 2\times1 + 2\times1 + \\ 3\times1 + 3\times1 + 3\times1 + \end{pmatrix} = 3+6+9 = 18$

Second Element $= 18$

Third Element $=$

2	2	2
3	3	3
4	4	4

$*$

1	1	1
1	1	1
1	1	1

$= \begin{pmatrix} 2\times1 + 2\times1 + 2\times1 \\ + 3\times1 + 3\times1 + 3\times1 \\ + 4\times1 + 4\times1 + 4\times1 \end{pmatrix} = 27$

Fourth Element $= 27$

So, the matrix will be

18	18
27	27

So, we have

$R =$ | | |
|---|---|
| 9 | 9 |
| 9 | 9 |

, $G =$ | | |
|---|---|
| 18 | 18 |
| 18 | 18 |

, $B =$ | | |
|---|---|
| 18 | 18 |
| 27 | 27 |

$\Rightarrow$ Element-wise addition of all 3-channels

$= \begin{bmatrix} 9+18+18 & 9+18+18 \\ 9+18+27 & 9+18+22 \end{bmatrix} = \begin{bmatrix} 45 & 45 \\ 54 & 54 \end{bmatrix}$

⇒ On Averaging each Element with $\frac{1}{9}$, The Conv Elements will be

$$= \begin{bmatrix} 5 & 5 \\ 6 & 6 \end{bmatrix}$$

b)

R =

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

*

1	1	1
1	1	1
1	1	1

∴ final Convolution matrix with zero padding

4	6	6	4
6	9	9	6
6	9	9	6
4	6	6	4

Channel G

0	0	0	0	0	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	0	0	0	0	0

*

1	1	1
1	1	1
1	1	1

⇒

final Convolution matrix with zero Padding

8	12	12	8
12	18	18	12
12	18	18	12
8	12	12	8

Channel B

final convolution matrix with zero padding

0	0	0	0	0	0
0	1	1	1	1	0
0	2	2	2	2	0
0	3	3	3	3	0
0	4	4	4	4	0
0	0	0	0	0	0

$*$

1	1	1
1	1	1
1	1	1

$\Rightarrow$

6	9	9	6
12	18	18	12
18	27	27	18
14	21	21	14

Element Wise addition of all channels

$=$

4	6	6	4
6	9	9	6
6	9	9	6
4	6	6	4

$+$

8	12	12	8
12	18	18	12
12	18	18	12
8	12	12	8

$+$

6	9	9	6
12	18	18	12
18	27	27	18
14	21	21	14

$=$

18	27	27	18
30	45	45	30
30	54	54	30
26	39	39	26

$\Rightarrow$ Resulting matrix after the convolution across 3 channels and Element wise addition.

In case of Averaging the result, the resulting matrix would be

$\Rightarrow \frac{1}{9}$

18	27	27	18
30	45	45	30
30	54	54	30
26	39	39	26

$=$

2	3	3	2
3.33	5	5	3.33
3.33	6	6	3.33
2.88	4.3	4.3	2.08

c) Dilated Kernel (with rate = 2)

$$K = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$ 5×5

R channel :- (with zero Padding)

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

1^{st} Element = 0+0+2+0+2

= 4

Similarly we need to compute for the rest of the matrix

Resulting matrix would be

4	4
4	4

G channel :-

0	0	0	0	0	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	0	0	0	0	0

$*$

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

1st Element = $4 + 4 = 8$

Similarly we need to Compute for the rest of the matrix and the feature Map will be =

8	8
8	8

B channel :-

0	0	0	0	0	0
0	1	1	1	1	0
0	2	2	2	2	0
0	3	3	3	3	0
0	4	4	4	4	0
0	0	0	0	0	0

$*$

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

Feature Map =

12	12
8	8

Element Wise addition of all channels

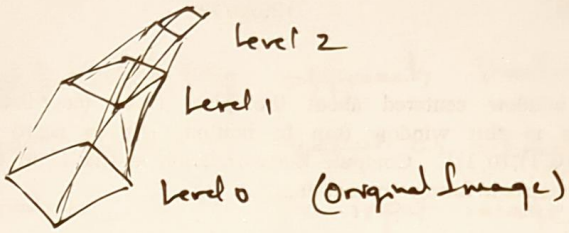
4	4	+	8	8	+	12	12	=	24	24
4	4		8	8		8	8		20	20

d) Template matching :-

Template matching approach involves defining a small image or Kernel, which is known as template. The template is then convolved with the input image by sliding it across the image, one pixel at a image and computing the similarity between the template and covered window on the image using the dot product.

It is implemented through 2D Convolution. In Convolution, the value of the o/p pixel is computed by multiplying elements of two matrices & summary the results.

e) Multiscale Analysis :-



To achieve the multiscale analysis with a fixed window size, a pyramid can be constructed by down-sampling the original image. By using the fixed size window at different scale, we can analyze the image at different scales and detect features more accurately.

f) In order to compensate the spatial resolution decrease, we can increase the depth or number of channels in a neural network to extract more complex features and compensate for the loss of information.

g) Input tensor = (128 x 128 x 32) ⇒ (h x W x channels)

Convolution filter size = (3 x 3) ⇒ (h x w)

$$\text{Output Width} = \frac{\text{Input Tensor width} - \text{filter width} + 2(\text{Padding})}{\text{Stride}} + 1$$

$$= \frac{128 - 3 + 2(0)}{1} + 1$$

$$= 125 + 1$$

$$= 126$$

Similarly, output height is also 126. Since there are 16 convolution filters, the channels in the o/p tensor would be 16.

output shape = (126 x 126 x 16)

h) $= \frac{128 - 3 + 2(0)}{2} + 1$

$= \frac{125}{2} + 1$

$= 62.5 + 1$

$= 63.5$

≈ 63

o/p tensor shape = $(63 \times 63 \times 16)$

i) The 1×1 Convolution layer can reduce the no. of channels in a feature map by using a 1×1 convolution with fewer filters.

ex:-

Input shape — 64×64×64

filter — 1×1×64, filters = 3

output shape — 64×64×3

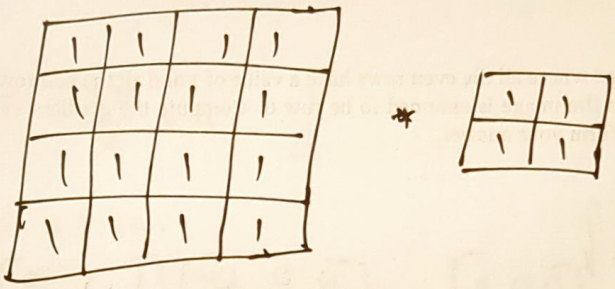
j) A Convolution layers generally used with ConvNet is a class of neural network that specializes in processing data that has a grid like topology such as an image

Convolution layers is the core building block of CNN. It performs the dot product b/w kernel and receptive field of image.

Difference b/w Early & deep layers :-

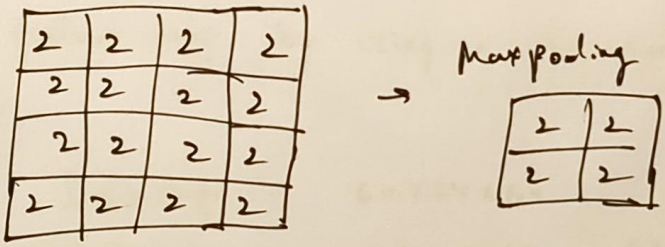
* Early Convolution layers can learn features like line, edges etc.
* Deep Convolution layers will learn more complex features such as face detection, object detection etc.

k) R channel :-

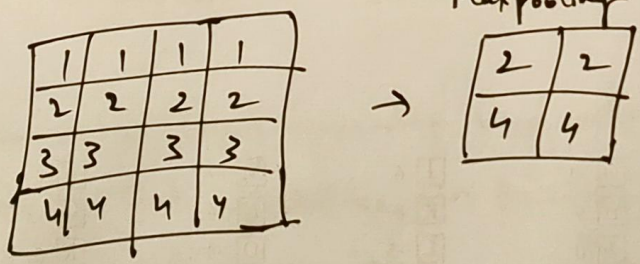


Max pooling with Stride 2 =

G channel :-



B channel :-



l) Pooling :- downsampling Spatial dimensions without changing the depth in the image.

Purpose of Pooling :-

1. It is used for Multiscale Analysis
2. Makes Convolutions Computationally less Expensive
3. Helps in reducing the no. of network Coefficients

m) Data Augmentation :- It is a technique which is used for increasing the size of a dataset by applying various transformations to the original data.

Purpose of data augmentation :-

* It helps the Machine learning model to improve the performance and generalization ability.
* It helps by adding more diversity to the training data which helps in reducing the overfitting.
* It is useful when training dataset is small.
* helps to address the class imbalance problem in the training data.

2. CNN

a) Transfer learning :- It is a technique in Machine learning where a pretrained model is used instead of training the model from scratch.

The purpose of transfer learning is to improve the performance and reduce the training time of a new model by using the knowledge and features learned from a pretrained model on some related task.

Transfer learning is useful when

* You have limited data available for training the model.
* the new task is related to that the pretrained model was trained on. In such cases, pretrained models learned features can be adopted to our task.
* new task requires high computational resources and we have limited time to train.

b) freezing the coefficients of the pretrained network refers to setting the parameters of a layer as non trainable to prevent them from being updated during the training procen.

The need for freezing is that pretrained models have learned features specific to the original task they were trained on, and freezing them will prevent the weights being disrupted during training on new task. This allows the new layers added to the network to be trained on the new task without affecting the pretrained layers learned features.

c) fine tuning a pretrained network involves adjusting its hyperparameters on a new dataset to improve its performance on a specific task. Steps included to fine tune a pretrained network-

i) Load a pretrained model using tensorflow or pytorch that you want to fine tune.

ii) Replace Last layer of the network with a new one that has the appropriate number of outputs for your new task.

iii) freeze some layers during the training in order to prevent them from updating. These layers belong to the pretrained network.

iv) After Configuring the network, train it on new dataset and Evaluate the performance of the fine tuned network on Validation set.

d) Inception Blocks :- Inception Blocks are a type of Building blocks used in CNN which are used to improve the ability to capture complex patterns in the input data. They use a combination of convolutional layers with different filter sizes and pooling operations to extract features at multiple scales and combine them to form a rich representation of the input.

To address the vanishing gradients problem in Googlenet, the network employed several solutions, including the use of the inception module, auxilary classifiers placed at intermediate layers, and the ReLU activation function.

The inception module allows the network to learn diverse features at different scales, while auxilary classifiers force the network to learn more meaningful representations. And ReLU helps prevent output from exploding and vanishing gradient problem.

e) Residual Blocks have skip connections that ~~pass~~ preserve the gradient signal during backpropagation that helps to prevent vanishing gradients problem. This allows the network to learn residual functions and focus on learning the fine grained details of the input. This is because the skip connection ensures that the gradient signal is preserved and not lost as it passes through the network and prevents it from vanishing gradient problems.

f) DenseNet :-

Each layer in the network is connected to the previous layer, which allows to reuse the features and reduces the parameters. DenseNet uses the Bottleneck like Onet to reduce the no. of featuremap before each dense block which helps to reduce the no. of parameters. Global Average pooling is used to reduce the no. of parameter and improve regularization. It does not use fully connected layer at the end of the nws. There is a transition layer in between each block which helps us to reduce the no. of feature maps and spatial dimensions.

g) let $R = \begin{bmatrix} 1&1&1&1 \\ 1&1&1&1 \\ 1&1&1&1 \\ 1&1&1&1 \end{bmatrix}$ $G = \begin{bmatrix} 2&2&2&2 \\ 2&2&2&2 \\ 2&2&2&2 \\ 2&2&2&2 \end{bmatrix}$ $B = \begin{bmatrix} 3&3&3&3 \\ 3&3&3&3 \\ 3&3&3&3 \\ 3&3&3&3 \end{bmatrix}$

$\text{filter1} = \begin{bmatrix} 1&1&1 \\ 1&1&1 \\ 1&1&1 \end{bmatrix}$ $\text{filter2} = \begin{bmatrix} 2&2&2 \\ 2&2&2 \\ 2&2&2 \end{bmatrix}$ $\text{filter3} = \begin{bmatrix} 3&3&3 \\ 3&3&3 \\ 3&3&3 \end{bmatrix}$

$R * \text{filter1} = \begin{bmatrix} 1&1&1&1 \\ 1&1&1&1 \\ 1&1&1&1 \\ 1&1&1&1 \end{bmatrix} * \begin{bmatrix} 1&1&1 \\ 1&1&1 \\ 1&1&1 \end{bmatrix} = \begin{bmatrix} 9&9 \\ 9&9 \end{bmatrix}$

$G * \text{filter2} = \begin{bmatrix} 2&2&2&2 \\ 2&2&2&2 \\ 2&2&2&2 \\ 2&2&2&2 \end{bmatrix} * \begin{bmatrix} 2&2&2 \\ 2&2&2 \\ 2&2&2 \end{bmatrix} = \begin{bmatrix} 36&36 \\ 36&36 \end{bmatrix}$

$B * \text{filter3} = \begin{bmatrix} 3&3&3&3 \\ 3&3&3&3 \\ 3&3&3&3 \end{bmatrix} * \begin{bmatrix} 3&3&3 \\ 3&3&3 \\ 3&3&3 \end{bmatrix} = \begin{bmatrix} 81&81 \\ 81&81 \end{bmatrix}$

* Both depthwise Conv & regular Conv yeilds same result.

h) MobileNets :-

i) Parameter Reduction :- MobileNets uses fewer parameters than other networks by using depthwise Separable Convolution and by reducing the no. of filters in each layer.

ii) MobileNets uses low resolution input which reduces the no. of computations required to process the image.

iii) Small filter Sizes :- MobileNets use small filter sizes (3×3) to reduce the no. of computations required by the Conv operation.

iv) Depthwise Separable Convolution :- MobileNets uses a depthwise Separable Convolution operation which splits a Standard Conv operation into two parts - depthwise Convolution & Pointwise Convolution which helps us to reduce the no. of computations while still achieving good accuracy.

3)

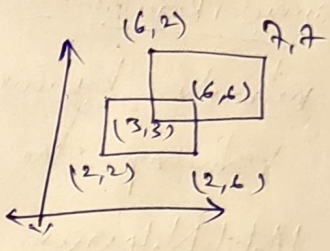
a) Tasks needed to be achieved during object detection.

i) object localization - Determine the spatial Coordinates or bounding box using regression

ii) object classification - To classify an object according to its class in the bounding box.
(i.e class labels)

b) deleted object (2,2) (6,6)
ground truth (3,3) (7,7)

$$IoU = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$



$|A|$ ↑ deleted object $|B|$ ↑ ground truth.

$|A| = 5 \times 5 = 25$ $|B| = 5 \times 5 = 25$

$|A \cap B| = (6-3)(6-3)$
$= 4 \times 4 = 16$

$\frac{16}{50-16} = \frac{16}{34} \frac{8}{17}$, Jaccard distance $= 1 - IoU$

$= 1 - \frac{8}{17} = 0.529$

c) To compute Avg precision we need to find calculate the precision & recall of each recall value. In this case, the unique recall values are 0.2, 0.4, 0.6, 0.8 & 1

Recall	precision
0.0	1.0
0.2	1.0
0.4	0.6
0.6	0.6
0.8	0.0
1.0	0.0

To compute AP 0.5 we need to integrate the precision - recall curve w.r to recall at IoU threshold of 0.5. This can be done using the trapezoidal Rule.

$AP(0.5) = (0.2-0)(1+1)/2 + (0.4-0.2)(1+1)/2 + (0.6-0.4)(0.6+1)/2 + (0.8-0.6)(0.6+0)/2$

$= 0.36$

d) Detection box of Coordinate :- Normalized between 0 & 1 boundaries ($x_{min}, y_{min}, x_{max}, y_{max}$). Since the image is at different scale, relative coordinates is used instead of absolute coordinates.

Scale invariance :- This would make the object detection algorithm scale invariant, which would detect objects equally well irrespective of the size.

e) Different terms in the loss function needed for an object detection network

$$loss = L_{reg} + L_{class}$$

L_{reg} :- loss due to regression of the bounding box

ex :- MSE

L_{class} :- loss due to classification of object inside the bounding box

ex :- Cross entropy

$$L(y^{(i)}, \hat{y}^{(i)}) = \begin{cases} (y_i - \hat{y}_i)^2 & \text{if } y_i = 0 \\ \varepsilon (y_i - \hat{y}_i)^2 & \text{otherwise} \end{cases}$$

Apart from these one other loss like object loss, box loss etc.

f) Grid cell of 3×3, size of the o/p tensor - 10 detected boxes at each cell

formula = $10 \times (x \times 5) * (k+5)$

$= 90 * (k+5)$

$= 90k + 450$

g) Single Shot vs double Shot

* Single Shot detector performs object detection in a single network that looks at image once.
* Double Shot detector look at the image twice - one to propose object regions which is referred an region proposal ~~detector~~ to refine and clarify region. This is more accurate but higher Computational Cost.

h) object detection Loss term :-

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \underbrace{\left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]}_{a \text{ — center regression loss}} +$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i}^{obj} \underbrace{\left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]}_{b \text{ — height \& weight regression loss}}$$

$$+ \sum_{i=0}^{S^2} \sum_{c} \mathbb{1}_{i}^{obj} \underbrace{(C_i - \hat{C}_i)^2}_{c \text{ — object classification loss}}$$

$$+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} \underbrace{(C_i - \hat{C}_i)^2}_{d \text{ — no object classification loss}}$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \underbrace{\sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2}_{e \text{ — box classification loss}}$$

i) RoI pooling :-

It extracts small feature map of fixed size. It follows a process of region proposal generation, where bounding boxes are generated by the network.

* Generates a RPN which represents the location of possible objects.
* ~~Does~~ for each & every subregion - pooling operation is applied reducing the spatial dimensions preserving the channel dimensions.

We do this because to provide a fixed size representation of variable sized image, which makes the object detection algo to handle objects of different sizes & different ratios. This helps in scale variation object detection & also helps in learning the feature of the object for different aspect ratio.

j) NMS (Non Maximum Supression)

RPN :- basically propose the region where the object has to be detected

Here non maximum supression is to select coordinate with highest ~~the~~ significance & delete candidate overlapping with it. (IOU > Threshold)

k) RCNN loss function

$$Loss = L_{class} + L_{reg} + L_{seg}$$

$$L_{class} = L^{Obj}_{class} + L^{RPN}_{class}$$

$$L_{reg} = L^{RPN}_{reg} + N^{Obj.}_{reg}$$

4.

a) Semantic Segmentation

In Semantic Segmentation it classifies each pixel into 1 or more classes. The goal is to group each and every pixel into specific class regardless of the object.

Instance Segmentation

In instance segmentation, it assigns unique label to each instance of the object and it provides accurate object boundaries with object localization.

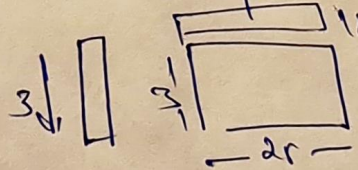
b) regular convolution

$$\text{o/p} :- \frac{n - \text{filter} + 2(\text{pad})}{\text{stride}} + 1$$

$$\frac{5-3}{1} + 1 = 2 + 1$$

$$= 3 \times 3$$

o/p :- 3x3 i/p :- 25x1



$3 \times 1 = 3 \times 25 \times 25 \times 1$

$4 = * \times$

∴ Size of the map = 3x25

of 3×25 $\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$

Size of the filter matrix = 3×25

(a) To perform Convolution b/w reshaped image & the filter, we need to slide the filter over the image. for image with dimensions 7×5 & filter size of 3×3, there are

$(5-3)+1 = 3$ possible positions to place the filter both horizontally & vertically. And so no. of operations that can be performed = 3×3 = 9 operations

Each Conv operations involves Convolving with 9×1 filter with the part of the region of 25×1 reshaped image that overlaps with the filter. This will produce the Convolution results of 9×9.

The final output will be 3×3 matrix where each pixel/element represents the results of the operation. (Assuming that padding or stride is used).

(c) To Compute the size of the transpose Convolution matrix, we need to reverse the Conv operation. Instead of sliding the filter over the image, we now slide the output of the Convolution over a larger matrix to obtain the original image. The output of Convolution will be 3×3 and results 5×5 by filling the gaps with zeros. Each element of the filter represents the subarea of the original image. We first fill in the corresponding position in transpose Convolution matrix with this output to get the transpose Convolution. There will be 9 possible positions in 5×5 matrix and so the transpose filter will be 5×5×9 with 9 channels. We need to flatten this now which results in (25×9).

d) Skip connection:-

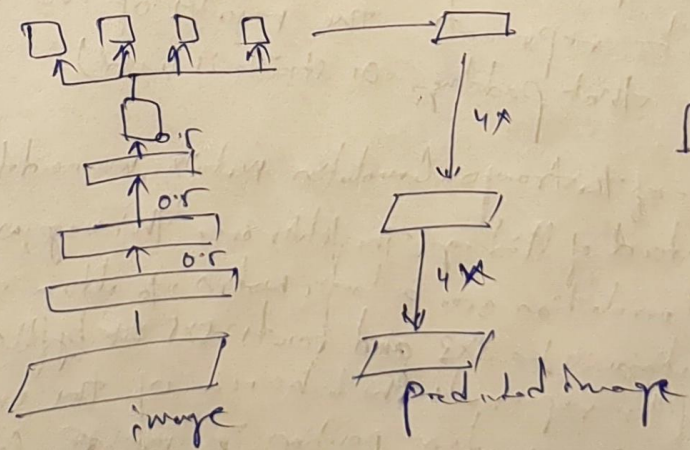
i) Connection b/w encoder & decoder to its corresponding layer (dimensions should be same.)

ii) Upsampled output is concatenated with cropped input - done due to loss of information (pixels) in every Convolutional layer)

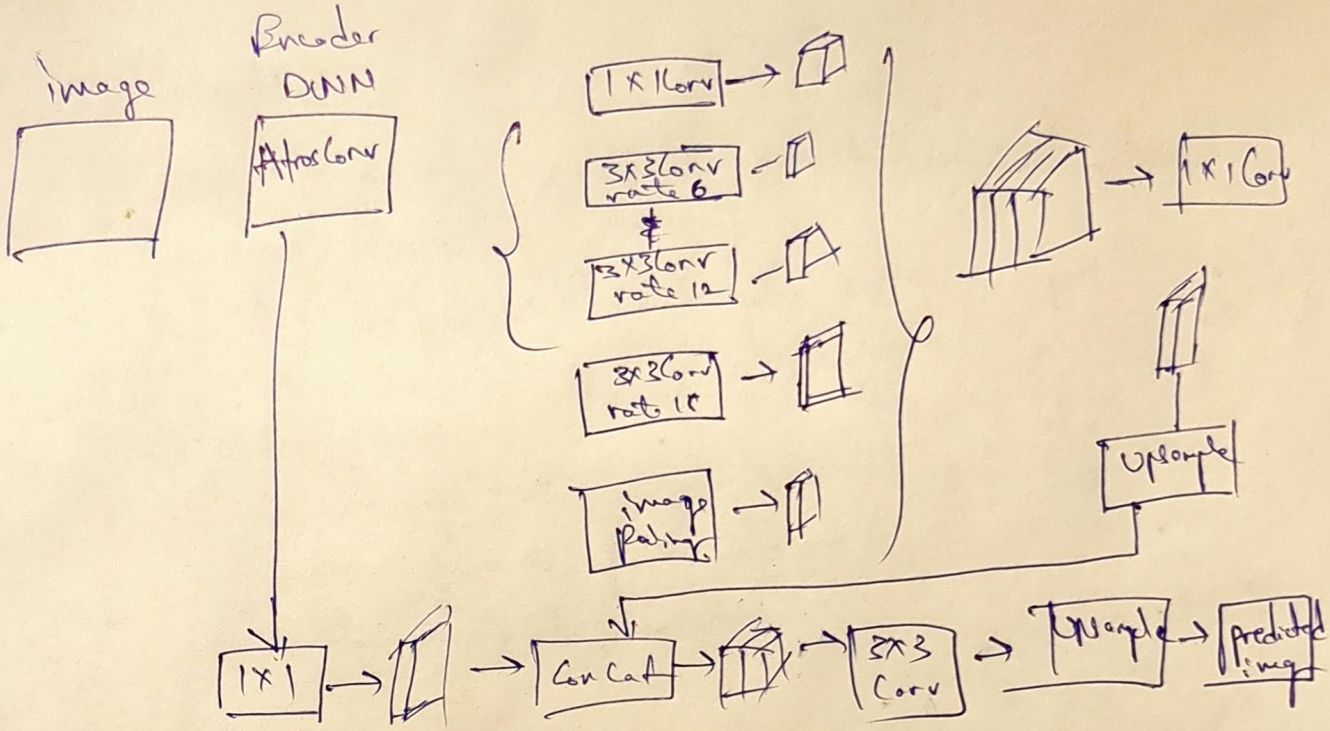
iii) Used to address vanishing gradient & improve the information flow & feature reuse which helps in accurate image segmentation.

e) Deeplab Architecture

* Encoder & decoder
* Uses Atrous spatial pyramid pooling with skip connection
* Uses depthwise separable convolution



Encoder-decoder Atrous Convolution



Multiscale Contextual information by applying atrous Convolution at multiple scale, the decoder defines the segmentation results along object boundaries.

f) Convolution matrices - Semantic segmentation results.

The possible evaluation matrices.

1. IoU
2. Pixel Accuracy :- no. of pixels that have been segmented into its corresponding class
3. Instances of Classification
   * precision
   * Recall
   * f1 Score.