Assignment - 4

Arinjay Jain
A20447307

Date __/__/__
Page

Theoretical questions →

(1)

R channel

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

G-Channel

2	2	2	2
2	2	2	2
2	2	2	2
2	2	2	2

B-Channel

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

2x3 filter

1	1	1
1	1	1
1	1	1

Result :

45	45
54	54

(2) with zero padding :

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

Red Channel →

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

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

filter is same as above.

Result →

18	27	27	18
30	45	45	30
36	54	54	36
26	37	37	26

(3) Using Dilated (atrous) convolution with dilation rate of 2.

A 3x3 kernel with a dilation rate of 2 will have the same field of view as a 5x5 kernel, while only using 9 parameters. Imagine taking a 5x5 kernel and deleting every second column and row.

Red Channel

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

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

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

Result: filter =

1	1	1
1	1	1
1	1	1

Result =

24	24
20	20

(4) Template matching interpretation of convolution :→
when ever a filter (f) convo with image (I) provide a response in a image form. we always want a higher (brigher) response.
we think of each filter applied to the image as a templete. when we convolve the image with that filter, we expect the portions of the image that match that particular templete to give a higher response.

☆ $f * I = R$ (response) will only higher if f is totally match with I . In this sense we are matching the templates using convolutions.

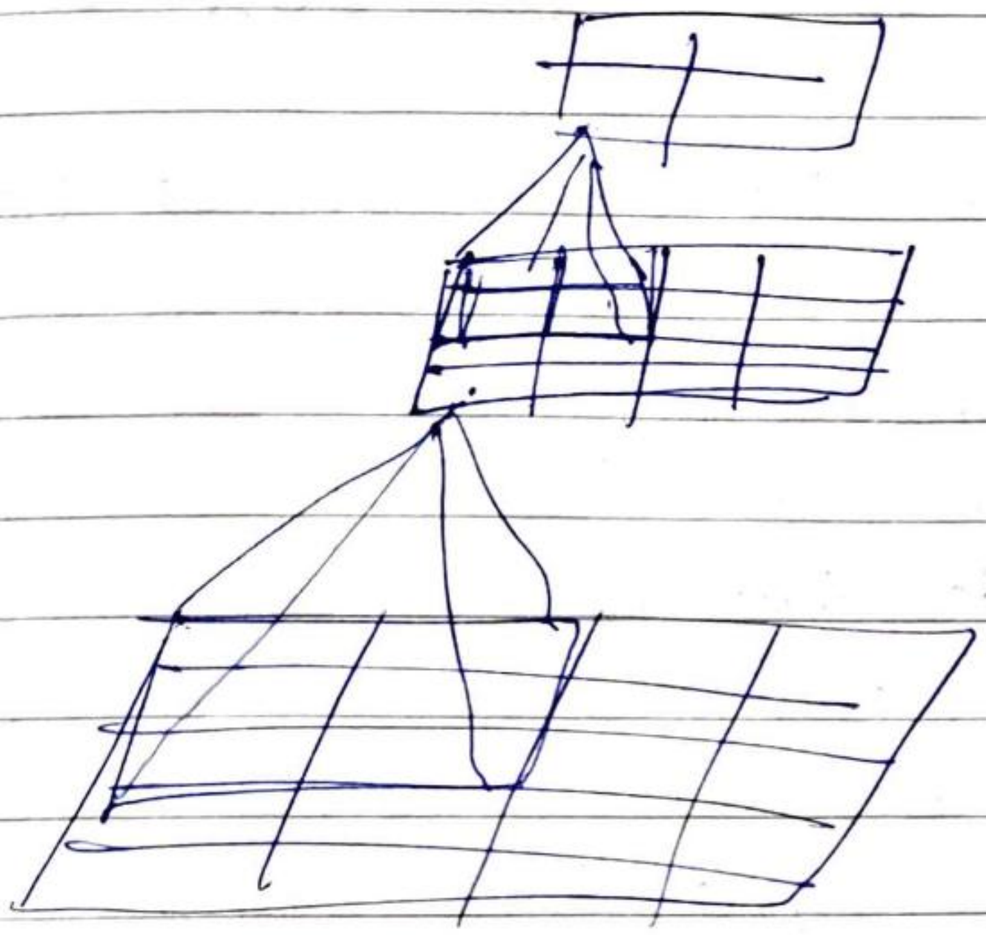
(5) with the fixed size of filter we can do multiple convolution on the same image at different scale. like we have 8x8 image and fixed filter of 2x2, after the 1st convolution we get an output of 4x4. after 2nd convolution with same filter we get 2x2.

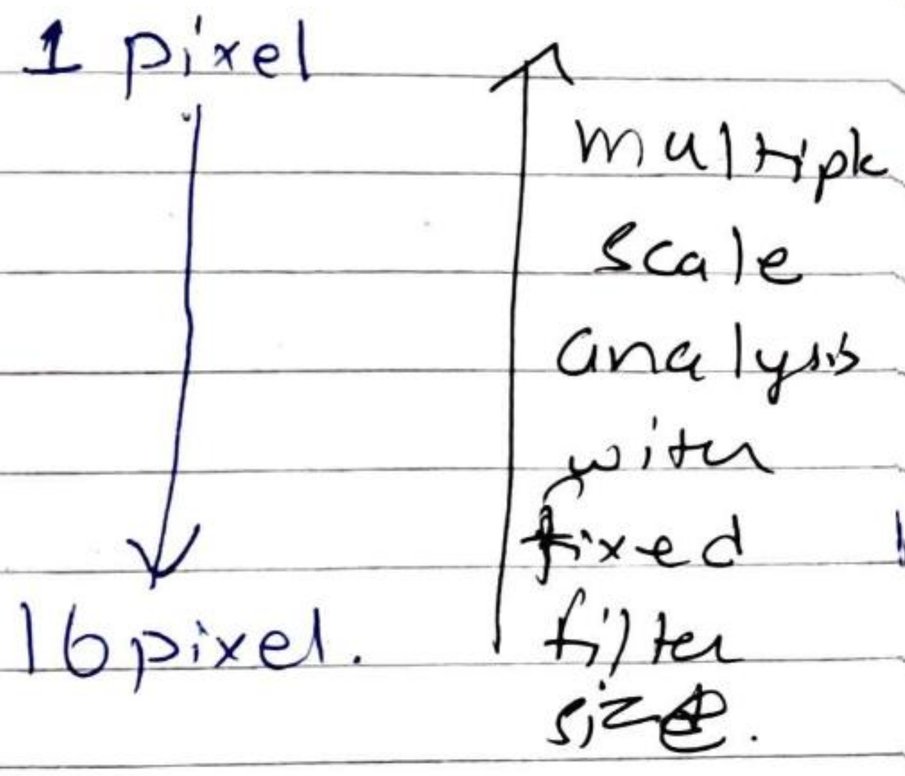
$$\underset{8\times8}{I} \times \underset{2\times2}{f} = \underset{4\times4}{O_1} \Rightarrow \underset{4\times4}{O_1} \times \underset{2\times2}{f} = \underset{2\times2}{O_2}$$

#

In this way, as we go higher in the pyramid of Convolutions, our receptive field in the convolved layer corresponds to a large deep receptive field in the original image and so it accounts for differently scaled objects.

⇒ multiple scale analysis.





(6) In each convolution layers, the size of the image reduces. This could lead to a loss of information as a result of a decrease in the spatial resolution. to compensat this we are adding the depth (no. of channel) in each convolution layers.

eg ⇒ Conv1 = 3 × 3 × 64 → number of Channel in Conv1 (Depth).

(7) $\left(\text{produce size} = \frac{W-K+2P+1}{S}\right)$

W → input volume, K → Kernel size; S → Stride
P → amount of zero padding.

Given ⇒ W = 128, P = 0, K = 3, S = 1 (16 - convo filters).

produce size = $\frac{128-3+0}{1} + 1 = 126.$

{ Size of resulting tensor = 126 × 126 × 16 }

(8) Same as previos, this time $S = 2$.

$$\text{Produce size} = \frac{\omega - k + 2p}{S_{//}} + 1$$

$$= \frac{128 - 3 + 2(0)}{2} + 1 = 63$$

{Size of resulting tensor = $63 \times 63 \times 16$.}

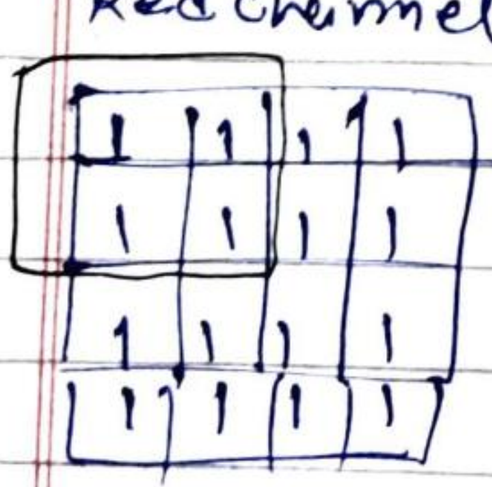
(9) 1×1 Convolution ⇒ the advantage of 1×1 Convolution filter is we can reduce the number of channel of given input without changing the image (input) size

eg ⇒ n - 1×1 filters through the image, where n is less than the current number of channels, we would get an output which is the same size as our input, but with 'n'-channels.

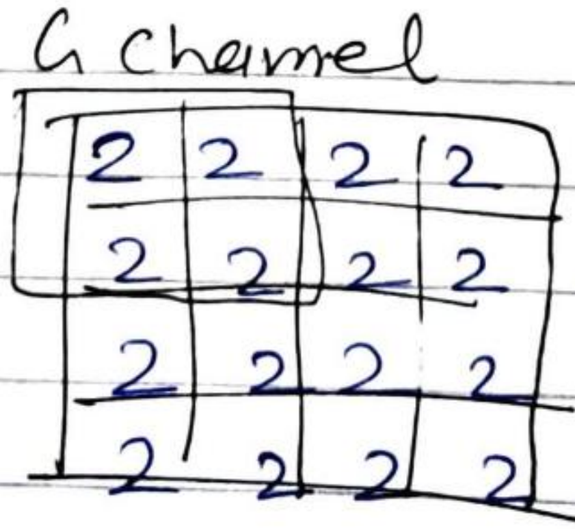
(10) The earlier layers in the convolutions are the Interpretation of convolution layers: is used to features extraction of input image.

The earlier layers in the convolutions are responsible for learning lower level features extraction in the input image. like as edges, while the deeper layers learn higher-level, more complex features of the images, like eyes, or ears or body parts in pictures of animals.

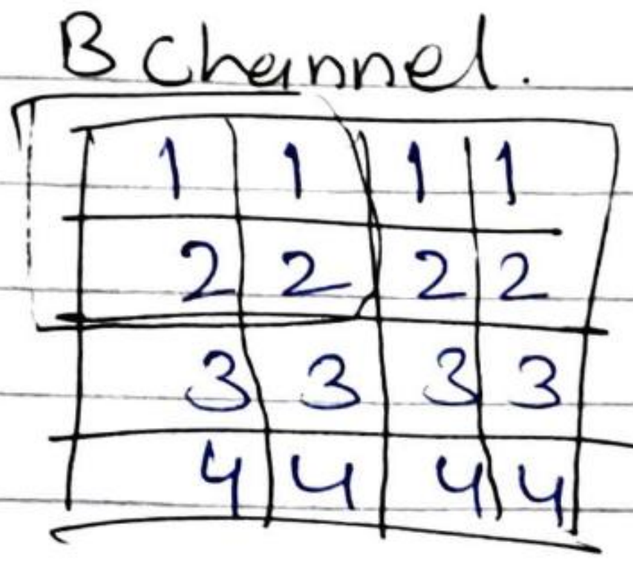
(11) Red channel



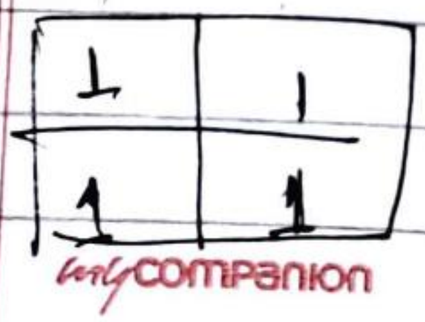
G channel

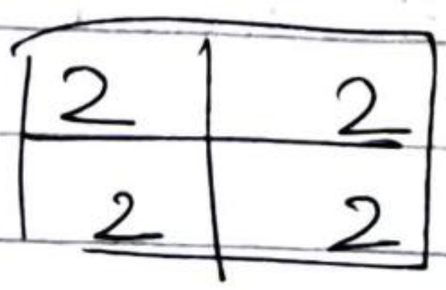


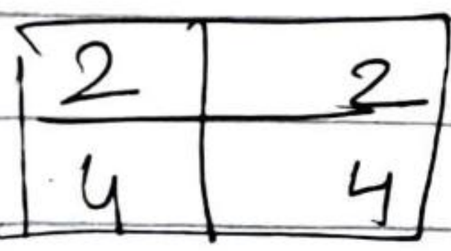
B channel.



Result after pooling.







(12) Purpose of pooling ⇒

Date __/__/__
Page

The purpose of pooling is the down sample the spatial images of the input image, without having to learn new parameters as in the case of convolutions.

(13) Data augmentation ⇒ we use data augmentation when we have limited data to use in model or network. it used to increas the dataset size, without actualy acquiring new data. eg ⇒ if we have a single image source so here we can create multiple images from original images by rotating or by moving in a fram at different location or shearing the corresponding image. Data augm-entation can help to avoid overfitting by increasing the dataset size.

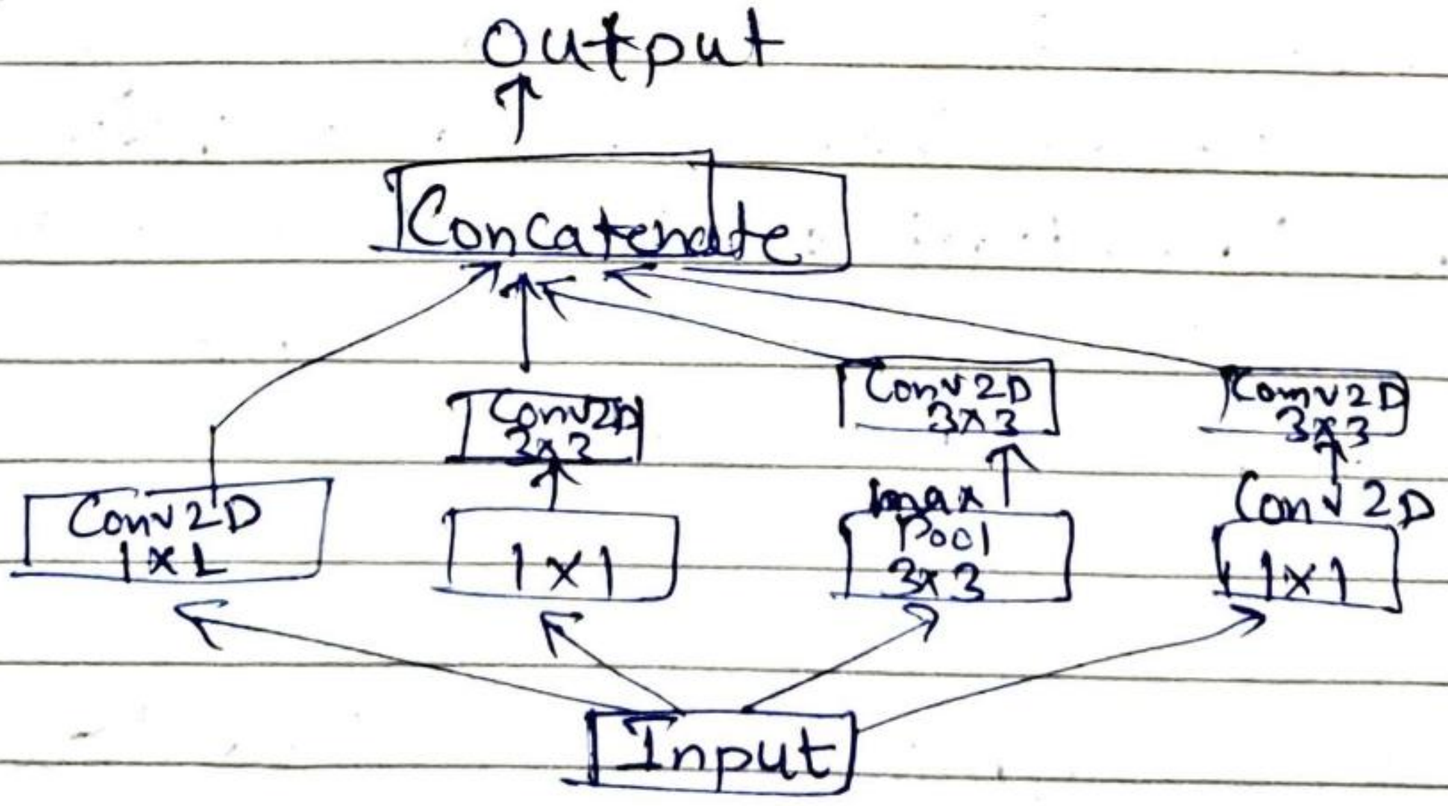
(14) Transfer learning. can be thought of as a feature extraction mechanism in the cases when we have a smaller training dataset at our disposal. by using a model that has been trained on a larger dataset, we use the pretrained model as an initial layer of our model to extract features. These extracted features can then be fed into a classifier that is customized for our use case.

(15) Freezing the coefficients of the pre-trained network. ⇒ we freez the weights of pre-trained layer that came from tranfer learning. because the weights are already trained and learnt. the top two layers of the model would be a custom classifier layers whose weights are initialized with small value ch close to zero. So when get a gradient value that, for the classifier layers, would be

small because the initial weights are small but the weights of the pre-trained layers are not small might be larger since they have already been learnt, which giver large gradient value. In this way we can destroy the weights that have already been learnt. to avoid this scenario the weights of the pre-trained model are frozen.

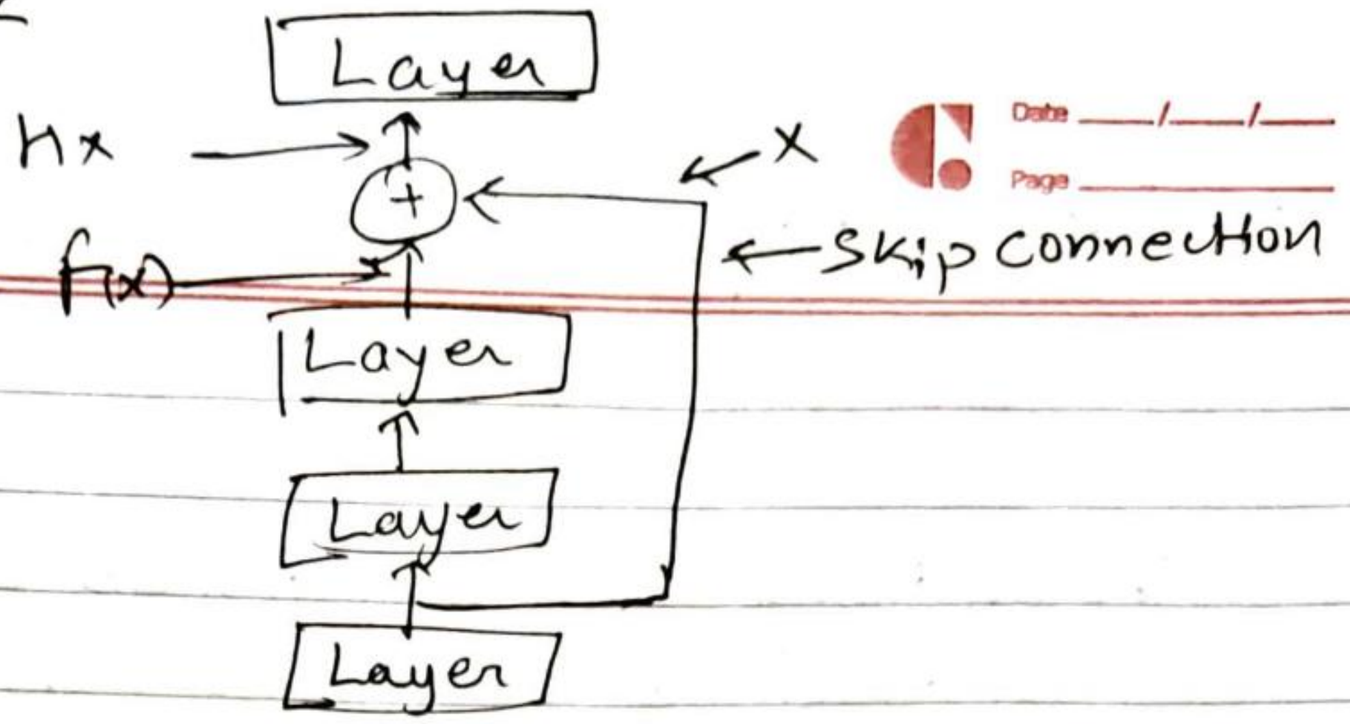
(16) After training our fully connected cal classifier layer we can un-freeze few of the top layers of the "Conv-base" pre-trained model, and retrain the entire model to fine tune the weights.

(17) Inception blocks ⇒ we convolve the ~~pre~~ previous layer with multiple filters, The outputs of these filters are then concotenated togethe. By doing so, we use more receptive fields, the concatenated outputs of which give us higher dimensionality in the output. Also, by passing a 1-D convolution before each Convolution block, we are able to reduce the number of parameters that need to be estimated.



(18) Residual block include a skip connection where an input layer x is added to the output of a subsequent layer. These skip connections help with vanishing gradients.

Residual block



(19) Once we have trained our convolution model we can create a new model that gives us multiple outputs - one output for each of the layers in our model. Plotting these individual outputs allows us to visualize the activations in each intermediate layer. Doing so allows us to visualize the templates that are learnt by the filters at each ~~inde~~ individual layer.

(20) To visualize the filter weights of the trained convolution layers, we can use gradient ascent to find the input that will maximize the response at the filter. Since we think of filters are templates that are being matched to the image, visualizing these filter ~~weight~~ weights in this way gives us a sense of the pattern or template that each filter looks for in the inputs.

(21) ⇒ To visualize the heatmap of class activation, we can follow the below algorithm ⇒

1. Feed an image to the network
2. Compute gradients of the selected output with respect to each channel of the target layer for which activations are to be computed, say the final convolution layer.
3. Compute the average gradients of each channel.
4. Add the activations of each channel weighted by their average gradient magnitude.

Date ___/___/___
Page _____

5. Superimpose the activations on the original image to get the heatmap.

Visualizing such a heatmap gives us a chance to understand which features of the original image influence the final classification output. Moreover heatmaps for misclassified images can point towards the problem points that have caused the error.