CS512 — AS4

Khalid Saifullah
A20423546

17th April 23
Spring Sem

2.(a) Transfer learning is a machine learning technique that uses a pre-trained model to improve the performance of a model on a new, similar task. It is useful when there is limited data, feature extraction is needed, domain adaptation is required, or faster training is desired. Transfer learning allows models to leverage knowledge from a source task to enhance performance on a target task, making it a powerful tool in various practical scenerios.

2 (b) Freezing the coefficients of a pre-trained network during fine tuning is important to retain the learned representations, prevent overfitting, stabilize training and improve computational efficiency. It allows the model to leverage the knowledge from the pre-trained model for the target task while avoiding drastic changes to the learned representations and making the fine-tuning process more efficient.

2 (c) Fine-tuning the coefficients of a pre-trained network involves updating the weights and biases of the pre-trained model while following these steps:

i) Remove or modify the output layer to match the target task.
ii) Freeze the pre-trained layers, except for the output layer, to retain the learned representations.
iii) Initialize the new layers, if added, with appropriate weights and biases.
iv) Train the model with target task data, updating the new layers while keeping the pre-trained layers frozen.
v) Monitor and adjust hyperparameters during training.

2 (d) Inception blocks are architectural modules used in CNNs to capture features at multiple scales or levels of abstraction using different receptive field sizes in a single layer. They were

introduced in GoogleNet to improve representational power and computational efficiency in deep networks. To address the issue of vanishing gradients, GoogleNet used bottleneck layers, which are 1×1 convolutional layers before 3×3 and 5×5 convolutions. These bottleneck layers reduce the number of input channels, reducing computational cost and mitigating vanishing gradients by introducing non-linearity. The combination of inception blocks and bottleneck layers in GoogleNet allows for efficient feature capture and gradient flow in deep networks.

2 (e) Residual blocks, also known as skip connections, are architectural modules used in deep neural networks to address the vanishing gradients problem. They allow for the bypassing of information from earlier layers to deeper layers, preserving gradient signals and facilitating more efficient weight updates during training. The addition operation in residual blocks combines the original feature representation with the updated features from the residual block. This helps to mitigate the vanishing gradients issue by ensuring that gradient signals do not become too small as they pass through the deep network, leading to improved training and convergence in very deep networks.

2 (f) Densenet is a neural network architecture that connects each layer to every other layer in a dense manner, promoting feature reuse and better gradient flow. It is constructed by stacking dense blocks with dense connectivity, and it controls complexity through the use of growth rate and compression factor. The growth rate determines the number of channels added to each layer, while the compression factor reduces the number of channels in the feature maps. These mechanisms allow for efficient feature representation, better gradient flow and control over model size and complexity.

2(h) MobileNets are designed to make computations faster on mobile and embedded devices through strategies such as depthwise separable convolutions, reduced model size, width multiplier, strided convolutions and pooling, and efficient network design. These techniques help reduce computation cost, memory requirements and redundant computations, making MobileNets well-suited for deployment on resource-constrained devices while maintaining good model performance.

3. (a) Object detection involves two main tasks: object classification which involves identifying and assigning class labels to objects in an image, and object localization, which involves accurately localizing objects using bounding boxes. These tasks are critical for object detection and enable a computer vision system to identify and locate objects within images, allowing for various applications such as object tracking, recognition and scene understanding.

3 (d) Detection box coordinates are normalized to be between 0 and 1 in object detection algorithms for several reasons, including scale invariance, simplified coordination, alignment with network output, and compatibility with common loss functions. Normalizing coordinates to this range allows for robustness to changes in object size, consistency in coordination system, easy alignment with network output, and compatibility with commonly used loss functions.

3 (e) The loss function in an object detection network typically includes several terms, including object classification loss, object localization loss, background classification loss and optional regularization loss.

3 (g) The main difference between single-shot and two-shot approaches in object detection is in the number of passes or iterations of the neural network used during training. Single-shot approaches use a single pass for object detection, while two-shot approaches involve multiple passes or iterations for region proposal generation, refinement, and classification. Single-shot approaches are generally faster but may have slightly lower accuracy, while two shot approaches may offer higher accuracy but may require more computation time due to additional iterations.

3(h) The YOLO object detector uses a specific loss function that consists of multiple terms:

i) Objectness Loss: This term measures the discrepancy between the predicted objectness ~~to~~ score and the ground truth for each anchor box.

ii) Classification Loss: Measures the difference between the predicted class probabilities and the ground truth class labels for each anchor box.

iii) Localization: Measures the difference between the predicted bounding box coordinates and the ground truth bounding box coordinates for each anchor box.

iv) Coordinate Loss: Measures the difference between the predicted coordinates of the bounding boxes and the ground truth coordinates.

v) ~~Cofin~~ Confidence: Measures the difference between the predicted confidence scores ~~(objectness score~~ and the ground truth confidence scores for each anchor box.

vi) Optional Regularization: This term is optional and can be added to the YOLO loss function to apply regularization on the model's parameters to prevent overfitting.

3(i) ROI pooling is a technique used in object detection networks to extract features from fixed-size regions of an input image corresponding to object proposals. It involves warping the input image to align with the network's feature maps, pooling features within proposed regions, resizing the pooled features and generating fixed-size features representations. ROI pooling helps the networks to extract informative features from different sized ~~features~~ object proposals and enables accurate object detection by aligning proposed regions with the network's spatial dimensions.

3(j) Non-maximum suppression is a post-processing technique used in

object detection to filter out redundant or overlapping bounding box predictions. It involves setting a confidence threshold, calculating the intersection over Union (IoU) between overlapping bounding boxes, and keeping only the most confident prediction among the redundant ones. NMS helps in generating a refined set of bounding box predictions and improving the accuracy of object detection tasks by eliminating duplicate detections.

3(k) The mask R-CNN loss function consists of three terms: RPN classification loss, RPN regression loss, and mask prediction loss. The RPN classification loss computes binary cross-entropy for classifying region proposals as foreground or background. The RPN regression loss calculates smooth L1 loss for refining the bounding box coordinates of the region proposals. The mask prediction loss computes binary cross-entropy for predicting pixel-wise object masks within the region of interest. These loss terms are combined and optimized during training to generate accurate region proposals, object detection, and mask prediction in the Mask-RCNN model.

4. (a) Semantic segmentation involves classifying each pixel in an image into predefined semantic classes, while instance segmentation goes a step further by not only classifying pixels but also differentiating instances of objects. Semantic segmentation does not distinguish between instances of the same object, while instance segmentation provides unique labels for each object instance in the image.

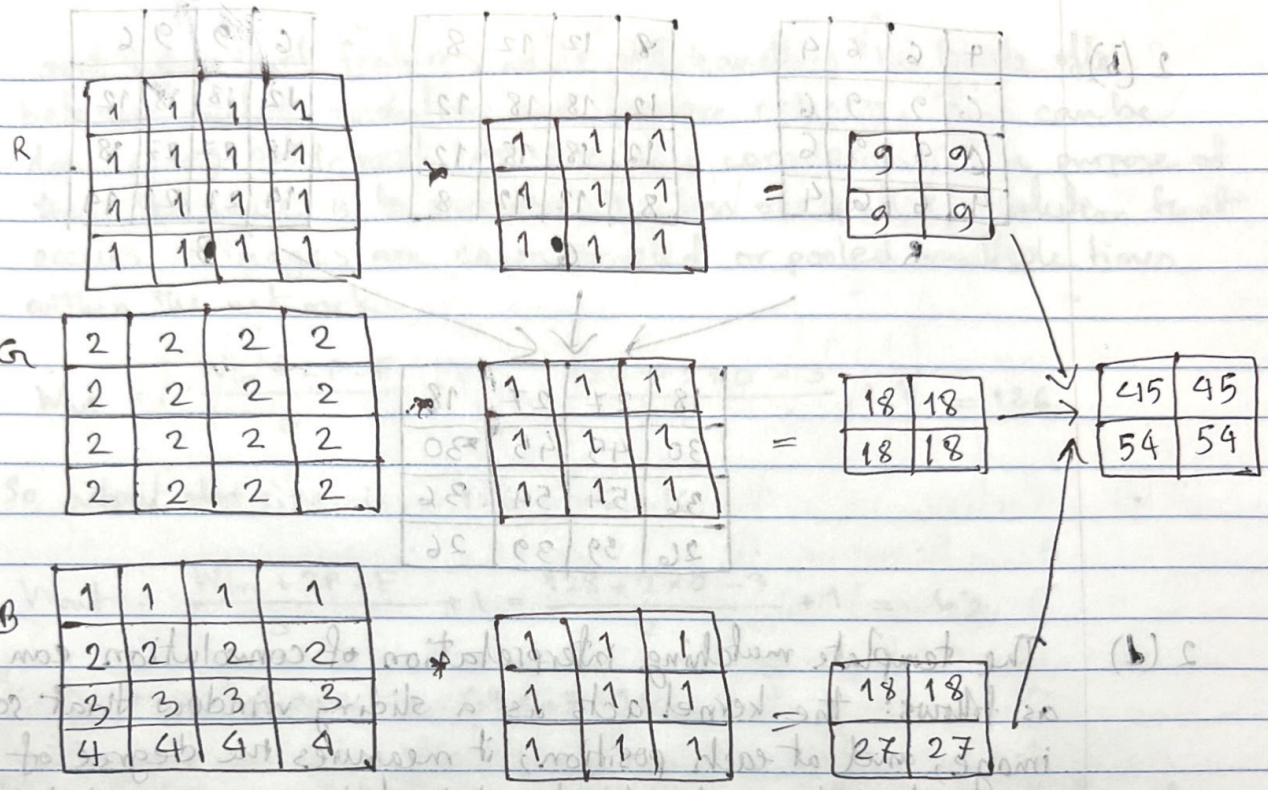
4 (d) Skip connections in U-Net are used to address the vanishing gradient problem and improve information flow between the encoder and decoder paths. They help to preserve spatial information during downsampling and facilitate accurate reconstruction of the original

image resolution in the decoder. Skip connections propagate information along the connections in a skip-and-fuse manner, allowing the decoder to recover spatial details while leveraging contextual information from the encoder.

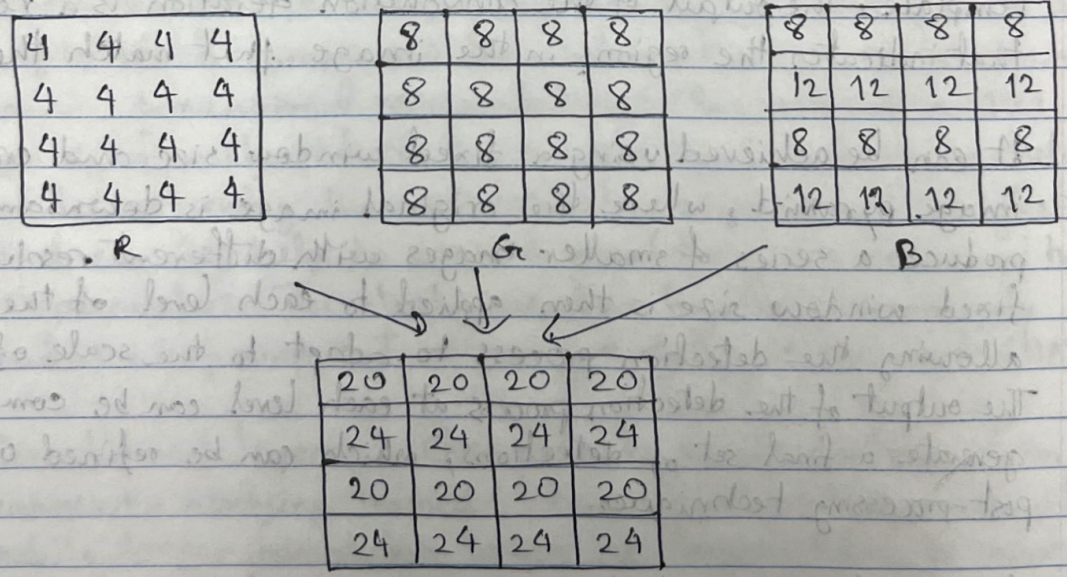
4(e) The DeepLab network architecture is a CNN used for semantic image segmentation. It consists of an encoder, atrous convolution, ASPP module, decoder, ~~fin~~ and final classification. Atrous convolutions and ASPP modules capture contextual information at multiple scale, while skip connections recover spatial details. The architecture allows for accurate and detailed segmentation results.

4(f) Intersection over Union (IoU) is a commonly used metric for evaluating semantic segmentation results. It measures the alignment between predicted and ground truth masks, with values ranging from 0 to 1. Higher IoU values indicate better segmentation accuracy, while lower values indicate poorer accuracy. IoU is widely used in semantic segmentation tasks to quantitatively evaluate model performance and compare different approaches. Other related metrics such as pixel accuracy, mean accuracy, and frequency weighted IoU may also be used in some cases.

2(a)



2(c) ~~(b)~~



2 (b)

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

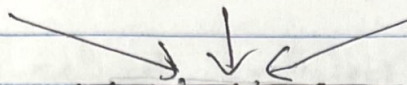
R

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

G

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

B



18	27	27	18
30	45	45	30
36	54	54	36
26	39	39	26

2 (d) The template matching interpretation of convolution can be explained as follows: the kernel acts as a sliding window that scans the image, and at each position, it measures the degree of similarity between the local patch of the image and the template. The output of the convolution operation is a response map that indicates the regions in the image that match the template.

2 (e) It can be achieved using a fixed window size and creating an image pyramid, where the original image is downsampled to produce a series of smaller images with different resolutions. The fixed window size is then applied to each level of the pyramid, allowing the detection process to adapt to the scale of the image. The output of the detection process at each level can be combined to generate a final set of detections, which can be refined using post-processing techniques.

2 (f) Compensating for spatial resolution decrease using depth is a technique used in computer vision to maintain or improve the accuracy of deep convolutional neural networks in capturing relevant visual features. This is achieved by increasing the depth of the feature maps, which allows the network to capture more complex patterns

and higher level features, while still handling the trade off between spatial resolution and feature richness. This can be done using bottle neck layers or skip connections. The purpose of this technique is to overcome the loss of spatial resolution that occurs as images are downsampled or pooled multiple times within the network.

2 (g) $W_{out} = \frac{W_{in} + 2P - F}{S} + 1 = \frac{128 + 2 \times 0 - 3}{1} + 1 = 126$

So, output ~~the~~ size is $126 \times 126 \times 16$

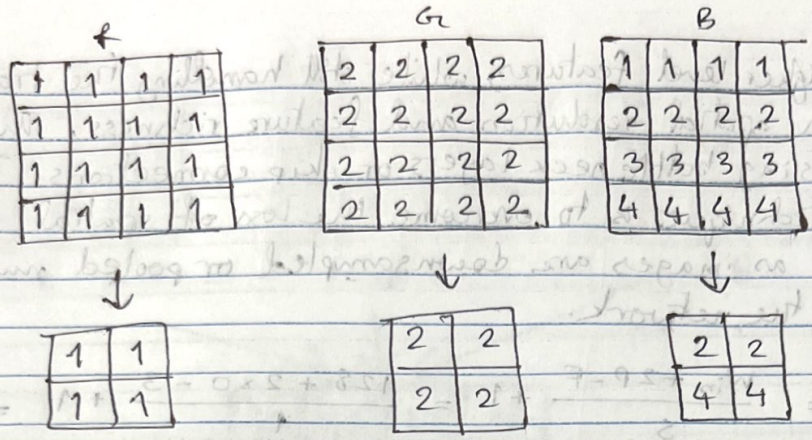
2 (h) $W_{out} = \frac{W_{in} + 2P - F}{S} + 1 = \frac{128 + 2 \times 0 - 3}{2} + 1 = 63$

So, the output shape is $63 \times 63 \times 16$

2 (i) Using a 1×1 convolution is a technique used in convolutional neural networks to reduce the number of channels in a feature maps, which compresses the information in each channel into a lower-dimensional representation. The use of a 1×1 convolution can significantly reduce the computational cost of the network while preserving the representational power of the feature maps and can also help to prevent overfitting by reducing the number of parameters. This technique has become increasingly popular and is a key component in many state-of-the-art CNN architecture.

2 (j) Convolutional layers in CNN extract visual features from input images by applying learned filters, which capture local patterns, while deeper convolution layers use larger filters to capture more complex and abstract features that build on the patterns learned in earlier layers. Deeper convolutions layers capture higher-level features that are more invariant to variations in the input image and are useful for tasks like object recognition and classification.

2 (k)

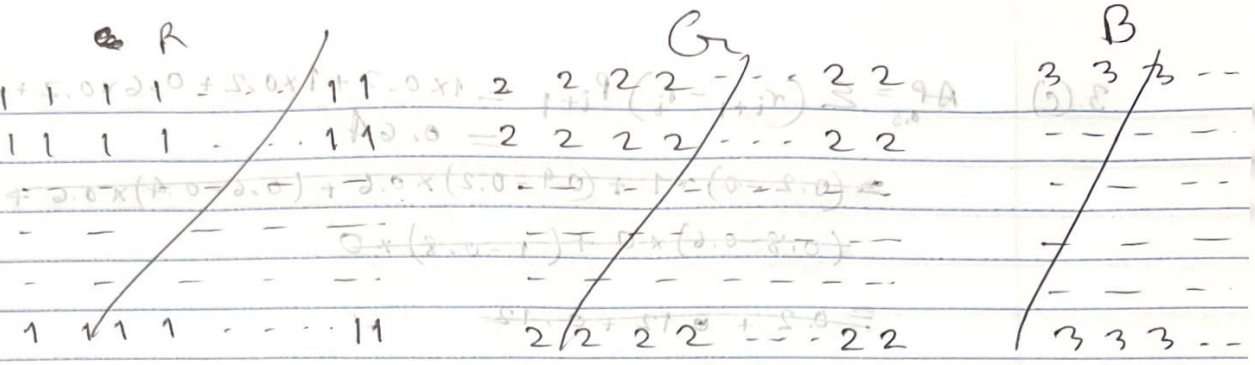


2 (l) Pooling is a technique used in machine learning and deep learning to reduce the dimensionality of feature maps. It involves taking the maximum, minimum or average value of non-overlapping regions in the input image or feature map. The purpose of pooling is to downsample the feature map, reduce the number of parameters in the network, prevent overfitting, and improve computational efficiency. Max pooling is the most commonly used type of pooling, as it tends to preserve the most important features and reduce the effect of noise in the input data.

2 (m) Data augmentation is a technique used in machine learning to increase the size of dataset by generating new data from existing data. Its purpose is to increase the diversity of the training data, prevent overfitting and improve the accuracy and robustness of the model. Data augmentation techniques include image flipping, rotation, cropping, zooming and color jittering and it is most useful when the dataset is small or imbalanced. Data augmentation is a powerful technique but needs to be applied carefully to avoid introducing unrealistic artifacts into the data.

4 (b) 17×25

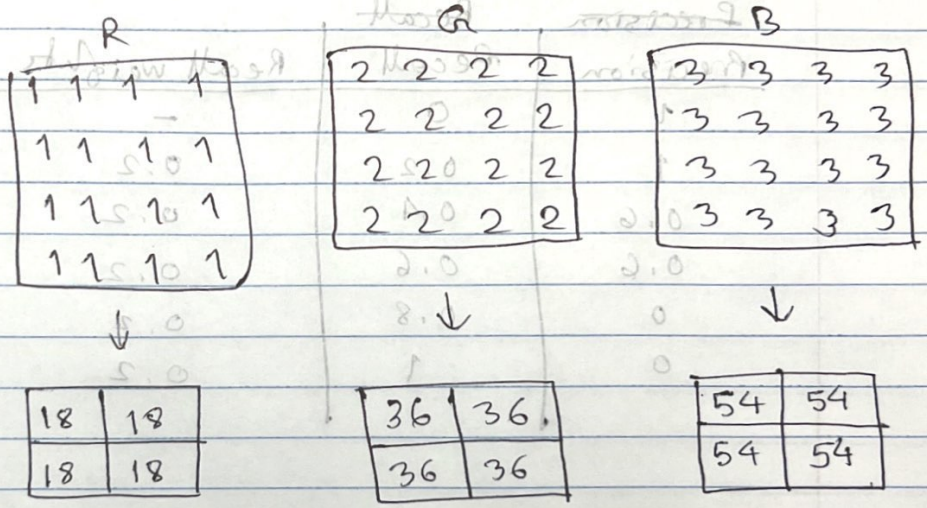
2.(g)



Filter

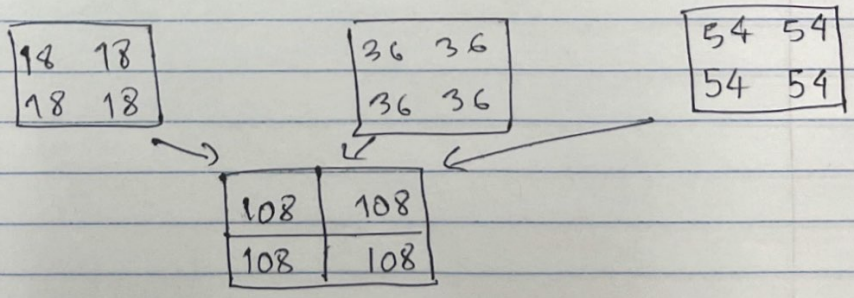
1	1	1
2	2	2
3	3	3

Depthwise Convolution



Pointwise Convolution

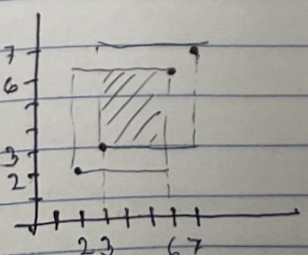
1	1
1	1



3(f) $y = [\underbrace{p_c}_{1}, \underbrace{x_c, y_c, w, h}_{4}, \underbrace{c_1, \dots c_k}_{10}]$

So, the size of tensor at each cell location is 1×15.

3(b)



$$IOU = \frac{4 \times 4}{5 \times 5 \times 2 - 4 \times 4} = \frac{16}{34} = \frac{8}{17} = 0.47$$

$$\text{Jaccard distance} = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} = \frac{34 - 16}{34}$$

$$= 0.53$$

3.(c) $AP_{0.5} = \sum (r_{i+1} - r_i) P_{i+1} = 1 \times 0.2 + 1 \times 0.2 + 0.6 \times 0.2 + 0.6 \times 0.2$

$= 0.64$

~~$= (0.2 - 0) \times 1 + (0.4 - 0.2) \times 0.6 + (0.6 - 0.4) \times 0.6 +$~~

~~$(0.8 - 0.6) \times 0 + (1 - 0.8) \times 0$~~

~~$= 0.2 + 0.12 + 0.12$~~

~~$= 0.26$~~

~~Precision~~ Precision	~~Recall~~ Recall	Recall weights
1	0	-
1	0.2	0.2
0.6	0.4	0.2
0.6	0.6	0.2
0	0.8	0.2
0	1	0.2