Week-11 PA DLP

1) Consider the following two blocks used in a U-Net architecture. Identify which one belongs to the encoder of the U-Net. **1 point**

A. Block A:

```
import torch

import torch.nn as nn

class BlockA(nn.Module):

    def __init__(self, in_channels, out_channels):

        super(BlockA, self).__init__()

        self.conv1 = nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1)

        self.relu = nn.ReLU()

        self.conv2 = nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1)

        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)

    def forward(self, x):

        x = self.conv1(x)

        x = self.relu(x)

        x = self.conv2(x)

        x = self.relu(x)

        x = self.pool(x)  # Reduces spatial dimensions

        return x
```

this is the task of encoder. downsampling module

B. Block B:

```
import torch

import torch.nn as nn

class BlockB(nn.Module):

    def __init__(self, in_channels, out_channels):

        super(BlockB, self).__init__()

        self.upconv = nn.ConvTranspose2d(in_channels, out_channels, kernel_size=2, stride=2)

        self.conv1 = nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1)

        self.relu = nn.ReLU()

        self.conv2 = nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1)

    def forward(self, x):

        x = self.upconv(x)  # Increases spatial dimensions

        x = self.conv1(x)

        x = self.relu(x)

        x = self.conv2(x)

        x = self.relu(x)

        return x
```

present in decoder

Which block belongs to the encoder of the U-Net?

- A. Block A ✓
- B. Block B
- C. Both
- D. Neither

which of the following correctly calculates the mse

A.

```
import torch

predictions = torch.tensor([2.5, 0.0, 2.0, 8.0])

targets = torch.tensor([3.0, -0.5, 2.0, 7.0])

mse = ((predictions - targets) ** 2).sum() / predictions.size(0)
```

B.

```
import torch

predictions = torch.tensor([2.5, 0.0, 2.0, 8.0])

targets = torch.tensor([3.0, -0.5, 2.0, 7.0])

mse = (predictions - targets).sum() / predictions.size(0)
```

C. ✓

```
import torch

predictions = torch.tensor([2.5, 0.0, 2.0, 8.0])

targets = torch.tensor([3.0, -0.5, 2.0, 7.0])

mse = ((predictions - targets) ** 2).mean()
```

$$\frac{1}{N}\sum_{i=1}^{N}(y^i - \hat{y}^i)^2$$

[(predic^{n} - target)^2].mean()

3) Consider the following code snippets for unsupervised depth estimation. Which one correctly incorporates left-right inconsistency into the loss computation? **1 point**

A. ✓

```
import torch

import torch.nn.functional as F

def left_right_loss(depth_left, depth_right, disparity_left):

    depth_reconstructed = F.grid_sample(depth_right, disparity_left)

    return torch.mean(torch.abs(depth_left - depth_reconstructed))
```

in the unsupervised depth estiman technique the process c/d LR inconsistency
↳ the loss funcⁿ used is absolute diff.

4) U-Net falls under which category of techniques- **1 point**

- A. Supervised ✓
- B. Unsupervised
- C. Semi-supervised

5) In depth estimation from stereo images, what is the primary role of the "disparity map"? → it will help you to compare L & R images and measure their displacement

1 point

- (a) To directly compute the depth values without requiring any other information.
- (b) To identify matching points between the left and right images and measure their displacement.
- (c) To smooth the textures in the images for better feature matching.
- (d) To estimate the camera's intrinsic parameters for depth calculation.

Yes, the answer is correct.
Score: 1

Accepted Answers:

(b) To identify matching points between the left and right images and measure their displacement.

6) What are the primary challenges of estimating a depth map from a single image using a multiscale deep neural network?

cues are not easily available, lack of global features

1 point

- (a) Understanding global context and handling variations in scene geometry and textureless regions.
- (b) Accurately aligning stereo image pairs and computing disparity for depth estimation.
- (c) Reducing computational complexity while matching points across two images.
- (d) Using depth sensors to directly capture depth information instead of estimating it.

7) What kind of pooling layer is used in Unet architecture?

1 point

- (a) Min pool of 2x2.
- (b) Max pool of 2x2.
- (c) Average pool of 2x2.
- (d) Average pool of 4x4.