

# DLP Week-12 PA

1) Which technique is utilized by LAKDNet to enhance feature extraction and aggregation?

1 point

- ☐ (a) Multi-Scale Convolutional Layers
- ☒ (b) Dual Attention Mechanism (Spatial and Channel Attention)
- ☐ (c) Recurrent Neural Networks (RNNs)
- ☐ (d) Depthwise Separable Convolutions

→ 4 layers

happens simultaneously

2) Suppose you want to add random noise to an image, where the noise values are uniformly distributed in the range [0, 20]. Which code snippet will correctly accomplish this?

1 point

(a)

```
import torch
```

```
def add_random_noise(image):
```

```
    noise = torch.rand_like(image) * 20 # Uniform noise in [0, 20]
```

```
    noisy_image = image + noise
```

```
    return noisy_image
```

3) What is the role of the **\*\*discriminator\*\*** in a Super-Resolution GAN (SRGAN)?

1 point

- ☐ (a) To generate high-resolution images from low-resolution inputs.
- ☐ (b) To compare generated images with ground truth and assign a similarity score.
- ☒ (c) To distinguish between real high-resolution images and generated high-resolution images.
- ☐ (d) To refine the generator's output by applying post-processing techniques.

SRGAN → G → generates artificial high-resolution images  
→ D → discriminates btw. real image and artificial images

Yes, the answer is correct.

Score: 1

Accepted Answers:

(c) To distinguish between real high-resolution images and generated high-resolution images.

4) What is a common challenge faced by CNN-based techniques for capturing fine textures in super-resolution tasks?

1 point

- ☒ (a) Difficulty in preserving high-frequency details like edges and textures, leading to overly smooth results.
- ☐ (b) Inability to generate accurate low-resolution representations for input images.
- ☐ (c) Overfitting to fine textures, resulting in loss of structural consistency.
- ☐ (d) Reduced performance due to high dependency on pre-trained models.

→ that's where DL comes to rescue

5) How many levels are there in the LAKDNet architecture?

1 point

- ☐ (a) 2
- ☐ (b) 3
- ☒ (c) 4
- ☐ (d) 5

(memory based)