

1. 다음은 PyTorch로 구현된 FCN 모델의 일부입니다. 아래의 코드 중 `self.fconn` 을 채워서 모델을 완성하세요.

```
class FCN(nn.Module):
    def __init__(self, num_classes=21):
        super().__init__()

        self.features = models.vgg16(pretrained=True).features

        self.fconn =

        self.score = nn.Conv2d(4096, num_classes, 1)

    def forward(self, x):
        features = self.features(x)
        fconn = self.fconn(features)
        score = self.score(fconn)

        return F.upsample(score, scale_factor=32, mode='bilinear', align_corners=True)
```

2. 다음은 MDNet으로 tracking하는 코드의 일부입니다. 제대로 실행하기 위해서 [1-4]에 어떠한 과정이 필요한지 서술하시오. (peseudo code)

1. [1] 현재 프레임에서 tracking할 box를 찾은 경우.
2. [2] 현재 프레임에서 tracking할 box를 찾지 못한 경우.
3. [3] short term update를 수행
4. [4] long term update를 수행

```
samples = sample_generator(init_bbox, opts.n_samples)
sample_scores = forward_samples(model, image, samples, opts, out_layer='fc6')

top_scores, top_idx = sample_scores[:, 1].topk(5)
top_idx = top_idx.cpu()
target_score = top_scores.mean()
init_bbox = samples[top_idx]
if top_idx.shape[0] > 1:
    init_bbox = init_bbox.mean(axis=0)
success = target_score > 0

if success:
    [1]
else:
    [2]

if not success:
    [3]
elif i % opts.long_interval == 0:
    [4]
```

