

# Computer Vision PA3 : Segmentation

20175051 Hoyeon Kim

Department of Physics and Photon Science  
Gwangju Institute of Science and Technology  
Gwangju, Korea

## I. INTRODUCTION

Segmentation은 이미지를 여러개의 파티션으로 나누는 작업입니다. PA3에서는 이미지를 foreground와 background로 나누는 binary segmentation을 진행합니다. SimpleOxfordPet-Dataset 데이터를 이용하여 Simple UNet, CSPN UNet (UNet with affinity), DSPN UNet (UNet with affinity and attention) 모델을 학습시키고 평가합니다. UNet 모델은 segmentation models pytorch pip 라이브러리[1]에 저장된 모델을 사용하였습니다. UNet의 weight는 기본값과 imagenet으로 pretrain된 weight 두 가지를 이용하였습니다. 코드의 대략적인 구현 방향은 아래와 같습니다.

- 1) Coarse segmentation을 얻기 위해 Simple UNet을 epoch 2번 학습시킵니다.
- 2) Simple UNet을 이용하여 coarse segmentation을 생성하고 이를 CSPN UNet에 넣어서 Affinity를 구합니다.
- 3) Affinity를 L1 norm으로 정규화하고 주변 값들을 1에서 빼서 Center Affinity를 구합니다.
- 4) Current segmentation을 3\*3 커널을 이용하여 Unfold한 뒤 중앙에 Coarse segmentation을 넣습니다.
- 5) 3과 4의 결과를 곱하고 이를 dim=1로 더해 current segmentation을 업데이트합니다. 이를 6번 반복합니다.
- 6) 2, 5를 반복하여 CSPN UNet을 epoch 40번 학습시킵니다.
- 7) 위 과정을 imagenet으로 pretrain된 weight를 initial하여 반복합니다.

## II. METHODOLOGY

Introduction에 소개된 프로세스를 구현하여 결과를 얻었습니다. PA3에서 사용된 모든 UNet의 Encoder는 resnet34, Encoder depth는 5로 동일합니다.

### A. Simple UNet

RGB image를 입력으로 Simple UNet에 넣어 Binary Segmentation된 Coarse segmentation을 얻습니다. Simple UNet의 입력 채널의 크기는 RGB = 3이고 출력 채널은 Binary = 1입니다.

### B. CSPN UNet

Simple UNet을 이용하여 Coarse Segmentation을 얻습니다. 이후에 RGB image와 Coarse Segmentation을 concat한 것을 입력으로 CSPN UNet에 넣어 Affinity를 얻습니다. CSPN UNet의 입력 채널의 크기는 RGB + Binary = 4이고 출력 채널은 Affinity = 8입니다. Affinity는 (batch size, 8, h, w)의 크기의 텐서입니다. 이를 F.normalize를 이용해 L1 distance, dim = 1로 정규화합니다. 그리고 1에서 normalized affinity 값의

합을 빼서 center affinity를 구합니다. 이를 적절하게 concat하여 affinity를 (batch size, 9, h\*w)의 크기로 변환합니다. 한편, current segmentation은 초기값을 coarse segmentation으로 시작하여 CSPN iteration = 6 동안 개선됩니다. 먼저 unfold를 이용해 3\*3 커널로 current segmentation을 unfold합니다. 이후에 가운데에 coarse segmentation을 넣습니다. unfold current segmentation은 (batch size, 9, h\*w)의 크기를 갖습니다. 마지막으로 affinity와 unfold current segmentation을 곱하여 output을 얻고 output을 dim = 1을 기준으로 더하여 current segmentation을 업데이트 합니다.

## III. RESULT

위에서 설명한 Simple UNet과 CSPN UNet의 학습 결과는 아래와 같습니다.

### A. Without imagenet pre-trained weight

Simple UNet을 epoch 2번 학습시킨 결과가 테이블 1과 그림 1에 나타나 있습니다. 그림 1에서는 RGB, GT, Coarse Segmentation이 나타나 있는데, 그림자가 있는 부분에서는 학습이 잘 안됩니다.

Metric	Training	Validation
Loss	1.6466	0.2426
Per Image IOU	0.6608	0.6252
Dataset IOU	0.6612	0.6252

TABLE I: Simple UNet training result, epoch = 2

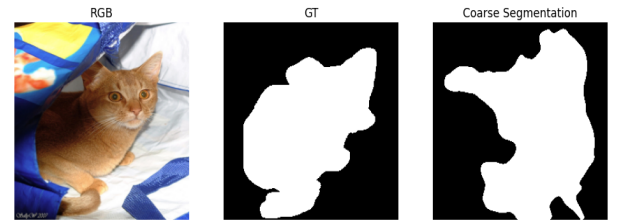


Fig. 1: RGB image, ground truth, and coarse segmentation of simple UNet

CSPN UNet을 epoch 40번 학습시킨 결과가 테이블 2과 그림 2, 3에 나타나 있습니다. 테이블 1, 2를 보면 Validation IoU는 Simple UNet에서 0.6252, CSPN UNet에서 0.7595로 CSPN을 사용하면서 증가하였습니다. 성능이 Simple UNet에 비하여 CSPN UNet에서 개선되었습니다. 그림 2를 보면 CSPN의 train loss와 validation loss는 epoch가 증가할수록 감소합니다. 모델이 overfit되지 않았습니다.

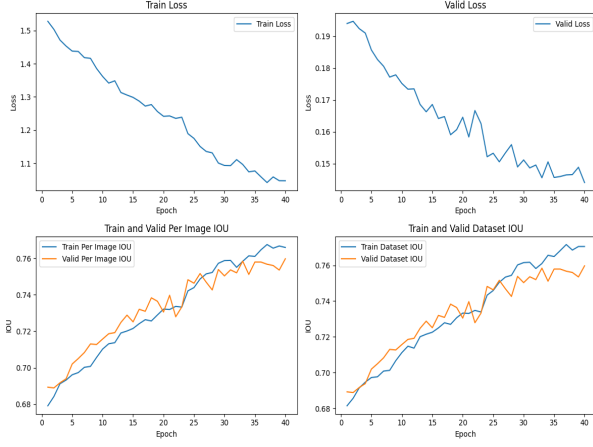


Fig. 2: CSPN train loss, validation loss and IoU

Metric	Training	Validation
Loss	1.0474	0.1441
Per Image IOU	0.7659	0.7597
Dataset IOU	0.7705	0.7597

TABLE II: CSPN UNet training result, epoch = 40

그림 3에서는 CSPN iteration별 Segmentation이 나타나 있는데, 그림자가 있는 부분에서는 학습이 잘 안됩니다. 우측 하단에 있는 Iteration 6이 Final Binary Segmentation입니다.

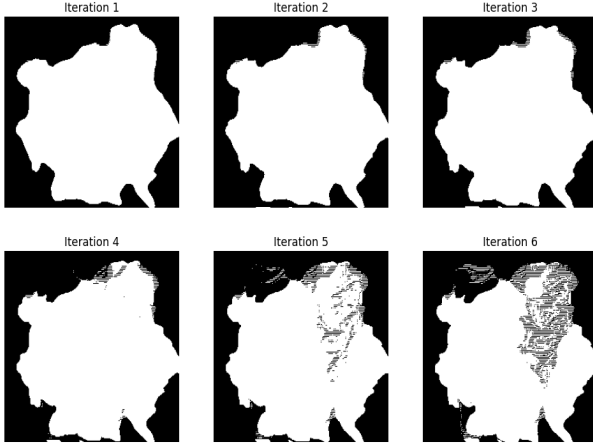


Fig. 3: CSPN segmentation of iteration 1 to 6

### B. With imagenet pre-trained weight

Simple UNet with imagenet pre-trained weight을 epoch 2번 학습시킨 결과가 테이블 3과 그림 4에 나타나 있습니다. 그림 4에서는 RGB, GT, Coarse Segmentation이 나타나 있는데, 그림자가 있는 부분에서도 학습이 되기 시작합니다.

Metric	Training	Validation
Loss	1.0093	0.1590
Per Image IOU	0.7778	0.7412
Dataset IOU	0.7785	0.7412

TABLE III: Simple UNet with imagenet weight training result, epoch = 2

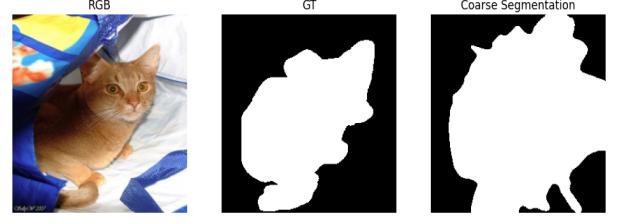


Fig. 4: RGB image, ground truth, and coarse segmentation of simple UNet with imagenet weight

CSPN UNet을 epoch 40번 학습시킨 결과가 테이블 4와 그림 5, 6에 나타나 있습니다. 테이블 3, 4를 보면 Validation IoU는 Simple UNet에서 0.7412, CSPN UNet에서 0.8237로 CSPN을 사용하면서 증가하였습니다. 성능이 Simple UNet에 비하여 CSPN UNet에서 개선되었습니다. 또한 pretrain된 weight를 사용하지 않을 때보다 사용할 때 성능이 개선되었습니다. 그림 5를 보면 CSPN의 train loss와 validation loss는 epoch가 증가할수록 감소합니다. 모델이 overfit되지 않았습 니다.

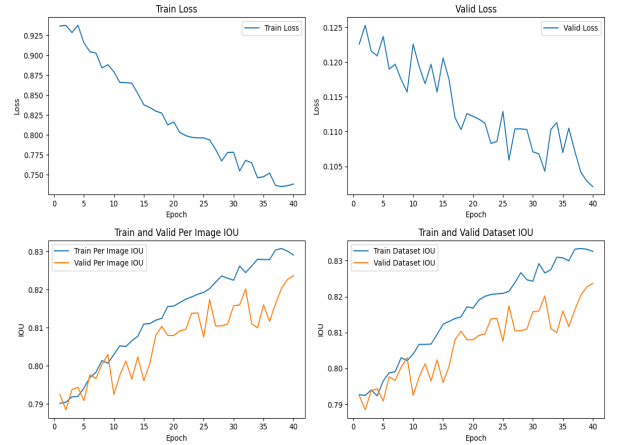


Fig. 5: CSPN with imagenet weight train loss, validation loss and IoU

Metric	Training	Validation
Loss	0.7384	0.1021
Per Image IOU	0.8291	0.8237
Dataset IOU	0.8326	0.8237

TABLE IV: CSPN UNet with imagenet weight training result, epoch = 40

그림 6에서는 CSPN iteration별 Segmentation이 나타나 있는데, 그림자가 있는 부분에서도 학습이 되기 시작합니다. 그러나 흰색 배경 위의 그림자를 아직도 잘 거르지 못하는 편입니다. 우측 하단에 있는 Iteration 6이 Final Binary Segmentation입니다. 또한 pretrain된 weight를 사용하지 않을 때보다 시각적으로도 더 잘 Binary Segmentation이 되었다는 것을 확인할 수 있습니다.

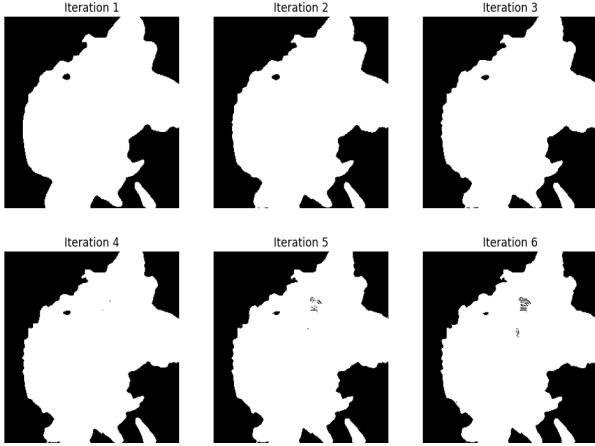


Fig. 6: CSPN segmentation with imagenet weight of iteration 1 to 6

#### IV. DISCUSS

Pre-trained weight를 사용하지 않으면 그림자가 있는 부분은 segmentation이 잘 되지 않습니다. 그런데 imagenet 데이터셋을 이용하여 pre-trained weight를 사용한 Unet은 그림자가 있는 부분에도 비교적 binary segmentation이 잘 동작합니다.

#### REFERENCES

- [1] Qubvel, "Segmentation Models Pytorch," GitHub repository. Available: [https://github.com/qubvel/segmentation\\_models.pytorch](https://github.com/qubvel/segmentation_models.pytorch).