고려대학교

DSBA Data Science & Business Analytics

ICNet for Real-Time Semantic Segmentation on High-Resolution Images

Zhao, H., Qi, X., Shen, X., Shi, J., & Jia, J.

(ECCV 2018)

고려대학교 산업경영공학과

DSBA 연구실 석사과정 이윤승

Content

1. Introduction

01) Several tasks in Computer Vision

(Dataset: Cityscapes)

[Semantic Segmentation]

- Pixel-wise classification
- 같은 class 물체가 있더라도,
  개별 object에 대한 고려 X
- Ex) PSPNet, ICNet

[Object Detection]

- Multiple object 위치 & class 분류
- 같은 class 물체도 다른 object 로 간주
- Ex) R-CNN, YOLO 계열 방법론

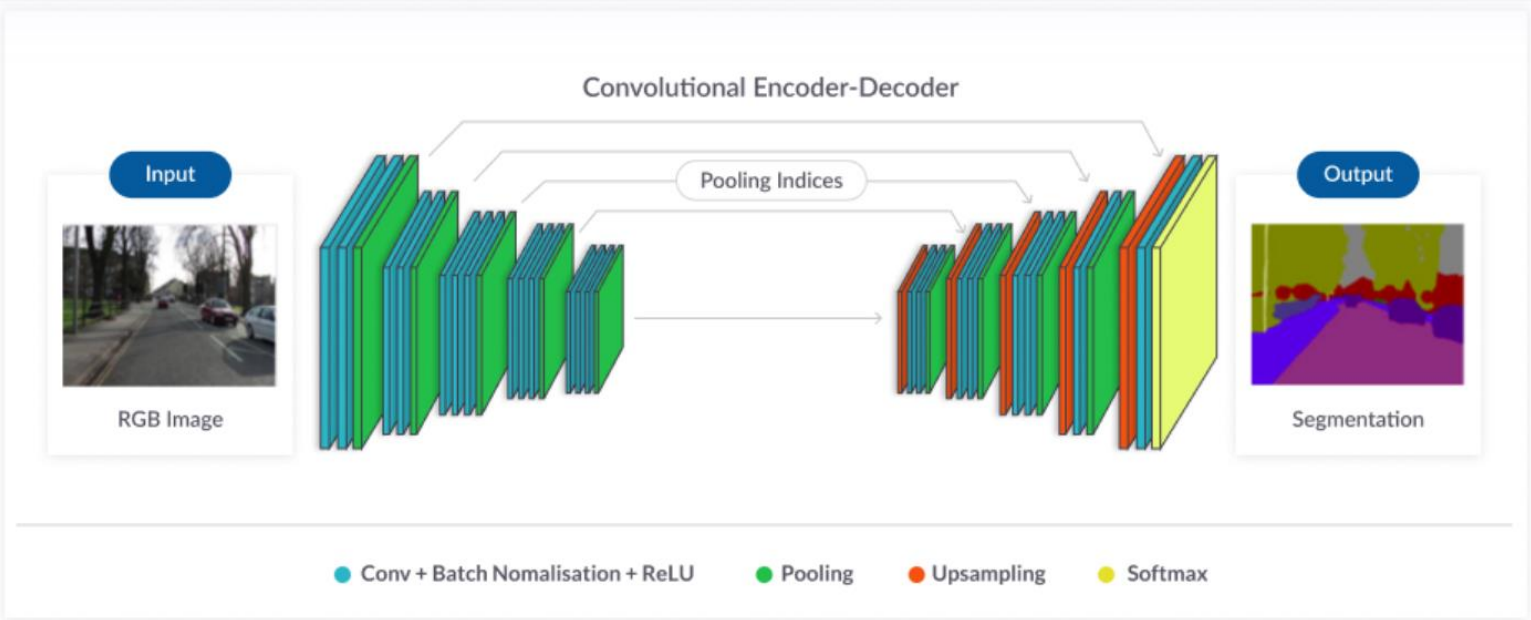
[Instance Segmentation]

- Multiple object 위치 & object 별 segmentation
- 같은 class 물체도 다른 object 로 간주
- Ex) Mask-RCNN (김동화 박사과정 발표영상)

1. Introduction

02) Semantic Segmentation

(Model Architecture: SegNet, U-Net, ENet)



✓ 대표적인 Semantic Segmentation 모델 구조 (Encoder-Decoder)

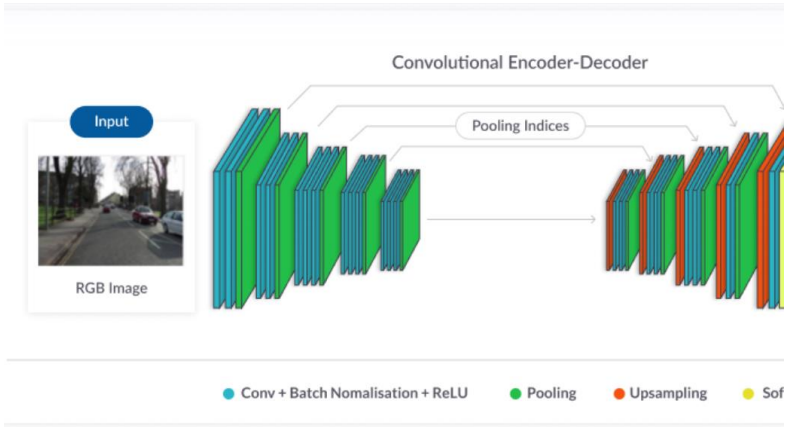
- 1단계) conv 연산을 통해 이미지 정보 축약 (encoder) 과정으로 얻은 feature map
- 2단계) 축소된 feature map 으로부터 upsampling 과 residual connection 통해 도출된 output tensor

  → 예측하고자 하는 Class 개수만큼의 channel 가짐

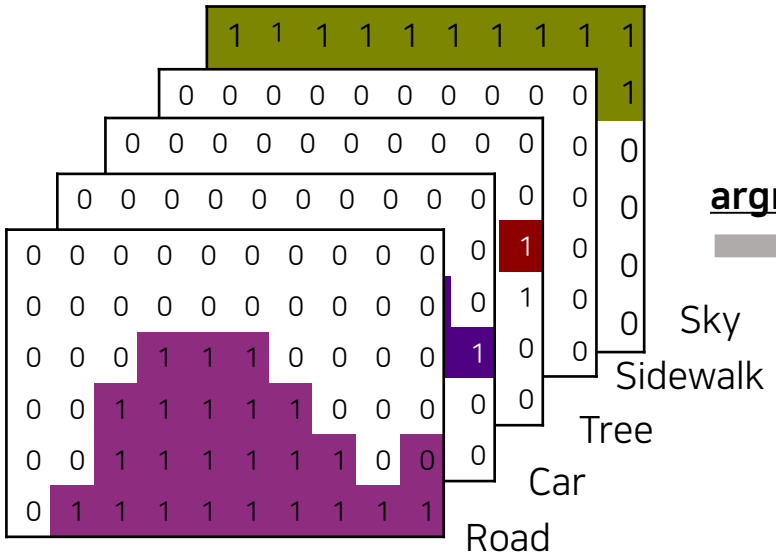
  → upsampling 기법: bilinear interpolation (*not learnable*), Transpose-convolution, dilated convolution

1. Introduction

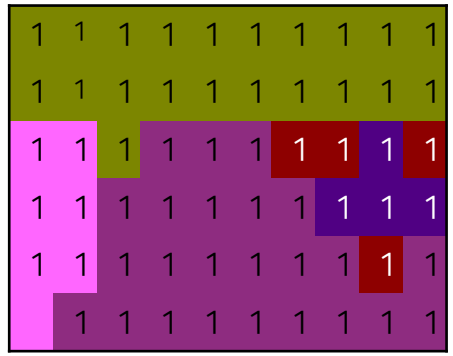
02) Semantic Segmentation

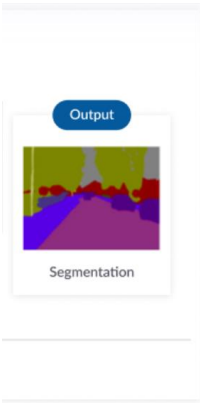


(value의 경우, 실제로는 softmax 값으로 표현됨)



argmax





Input Image
W x H x 3

Output Tensor
W x H x 5

Final Prediction
W x H x 1

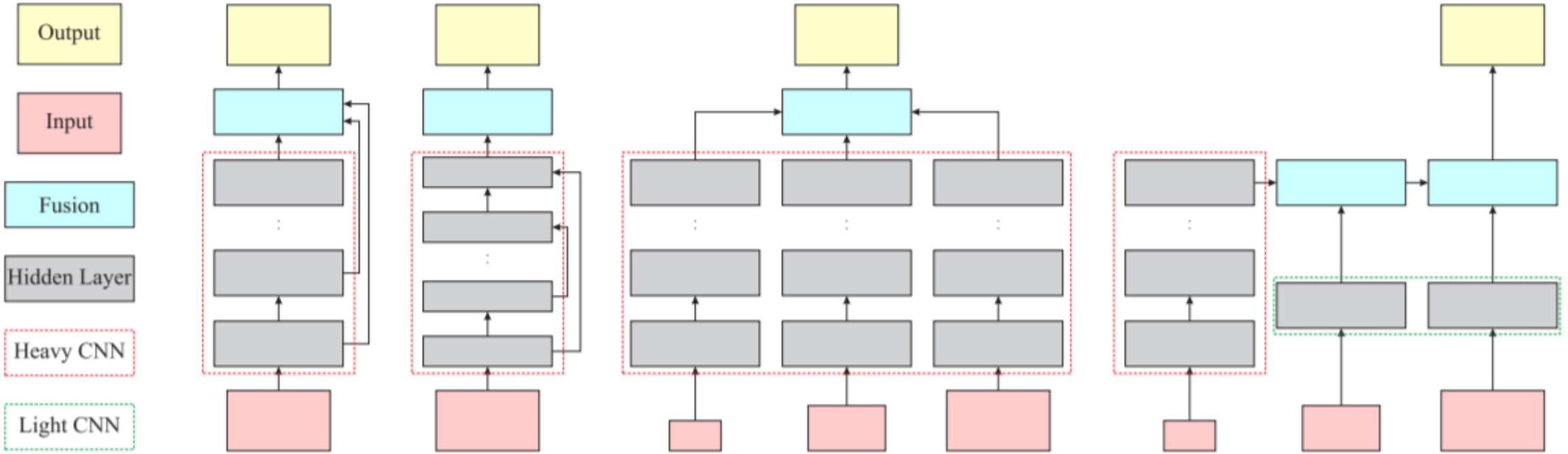
- 3단계) argmax 연산을 통해 해당 위치의 pixel 에서 가장 softmax 값이 높은 class 로 예측수행

  → loss function: categorical cross entropy

  → GT label: 각 pixel 별 class 정보가 있어야 함.

1. Introduction

02) Semantic Segmentation



모델	FCN	SegNet, UNet	DeepLab, PSPNet	ICNet
특징	Skip Connection	Encoder-Decoder	Multi-scale prediction Ensemble	Only Low Resolution used as input to heavy CNN

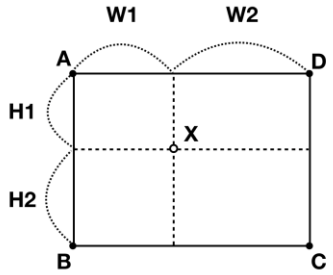
1. Introduction

03) Upsampling Technique

- 사용이유: CNN 거쳐 나온 feature map은 coarse 하므로 pixel-wise prediction 위한 dense feature map을 얻기 위해

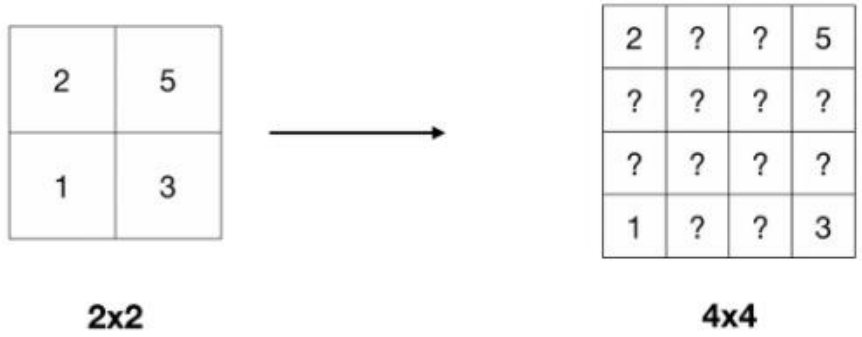
[Bilinear Interpolation]

- 기본 원리



$$X = \left(A\frac{H2}{H1+H2} + B\frac{H1}{H1+H2}\right)\frac{W2}{W1+W2} + \left(D\frac{H2}{H1+H2} + C\frac{H1}{H1+H2}\right)\frac{W1}{W1+W2}$$

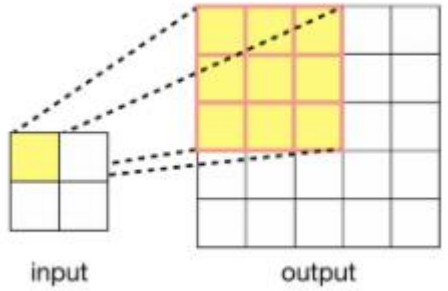
- Feature map 적용 (*not learnable*)



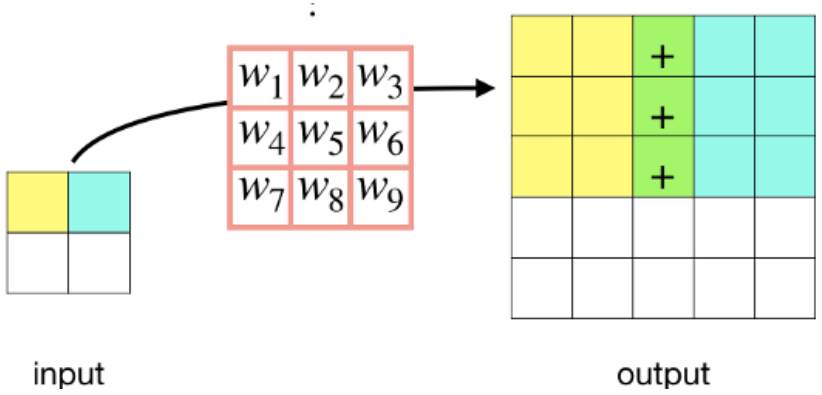
[Transpose Convolution]

- 기본 원리

ConvTranspose2d



- Feature map 적용 (*learnable*)



1. Introduction

03) Upsampling Technique

- 사용이유: CNN 거쳐 나온 feature map은 coarse 하므로 pixel-wise prediction 위한 dense feature map을 얻기 위해

[Dilated Convolution]

- 기본 원리

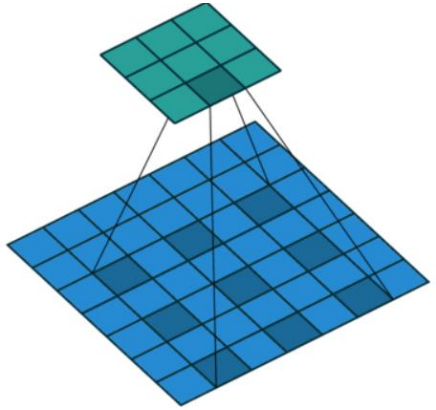
- Conv2d (option: dilation)

- Filter 내부에 zero padding 추가해 receptive field 확장

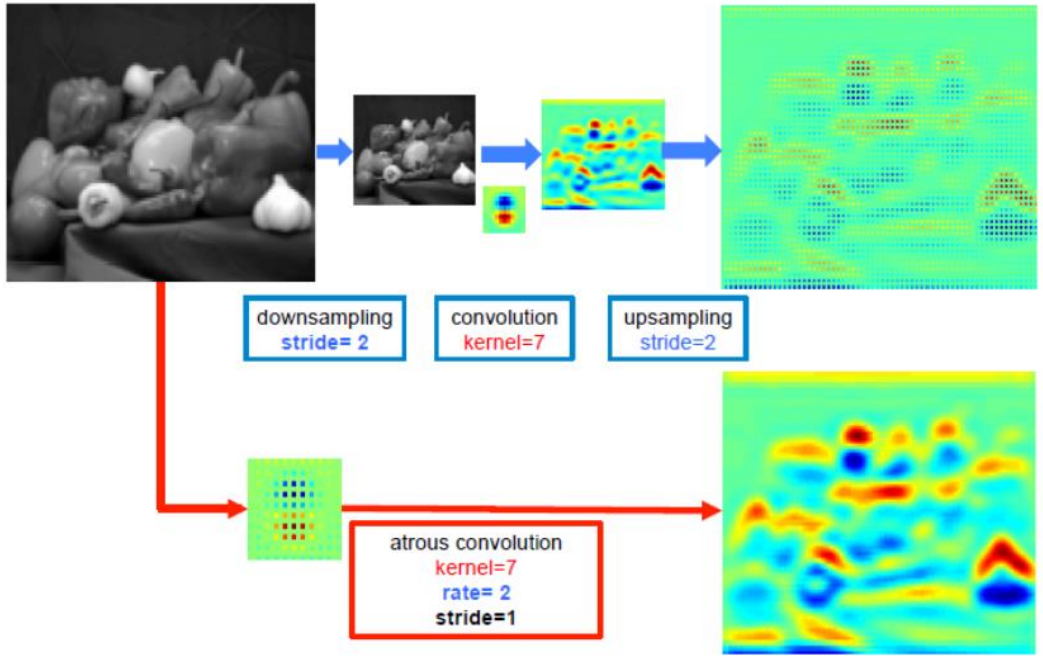
- 넓은 Receptive field 유지하며 적은 파라미터로 학습 가능

⇒ global context 보존 위해

- Feature map 적용 (*learnable*)



- 예시



1. Introduction

[1] PSP-Net (CVPR, 2017)

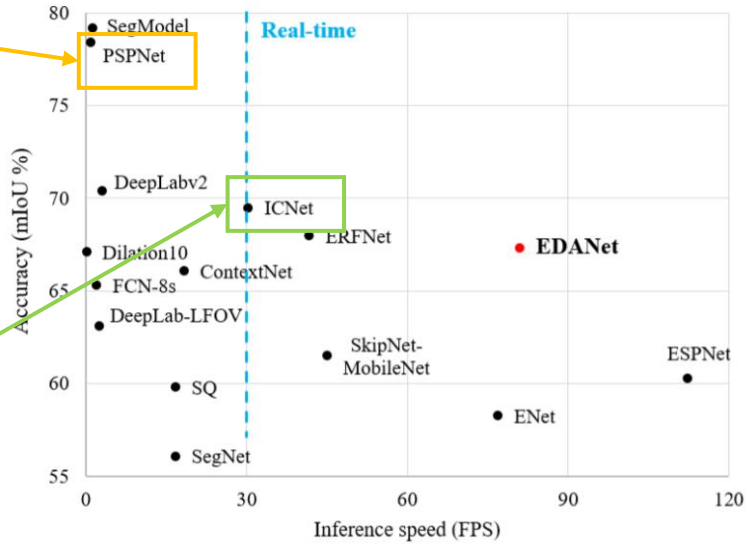
Pyramid Scene Parsing Network

Hengshuang Zhao[1] Jianping Shi[2] Xiaojuan Qi[1] Xiaogang Wang[1] Jiaya Jia[1]
[1]The Chinese University of Hong Kong [2]SenseTime Group Limited
{hszhao, xjqi, leojia}@cse.cuhk.edu.hk, xgwang@ee.cuhk.edu.hk, shijianping@sensetime.com

[2] IC-Net (ECCV, 2018)

ICNet for Real-Time Semantic Segmentation on High-Resolution Images

Hengshuang Zhao[1], Xiaojuan Qi[1], Xiaoyong Shen[2], Jianping Shi[3], Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong, [2] Tencent Youtu Lab, [3]SenseTime Research
{hszhao,xjqi,leojia}@cse.cuhk.edu.hk,
dylanshen@tencent.com, shijianping@sensetime.com



1. Introduction

[1] PSP-Net (CVPR, 2017)

Pyramid Scene Parsing Network

Hengshuang Zhao[1] Jianping Shi[2] Xiaojuan Qi[1] Xiaogang Wang[1] Jiaya Jia[1]
[1]The Chinese University of Hong Kong [2]SenseTime Group Limited
{hszhao, xjqi, leojia}@cse.cuhk.edu.hk, xgwang@ee.cuhk.edu.hk, shijianping@sensetime.com

[PSP-Net 특징]

- 81.2% mIOU 로 정확도 매우 높음
- 0.78 fps 로 매우 느린 모델에 속함

[2] IC-Net (ECCV, 2018)

ICNet for Real-Time Semantic Segmentation on High-Resolution Images

Hengshuang Zhao[1], Xiaojuan Qi[1], Xiaoyong Shen[2], Jianping Shi[3], Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong, [2] Tencent Youtu Lab, [3]SenseTime Research
{hszhao,xjqi,leojia}@cse.cuhk.edu.hk,
dylanshen@tencent.com, shijianping@sensetime.com

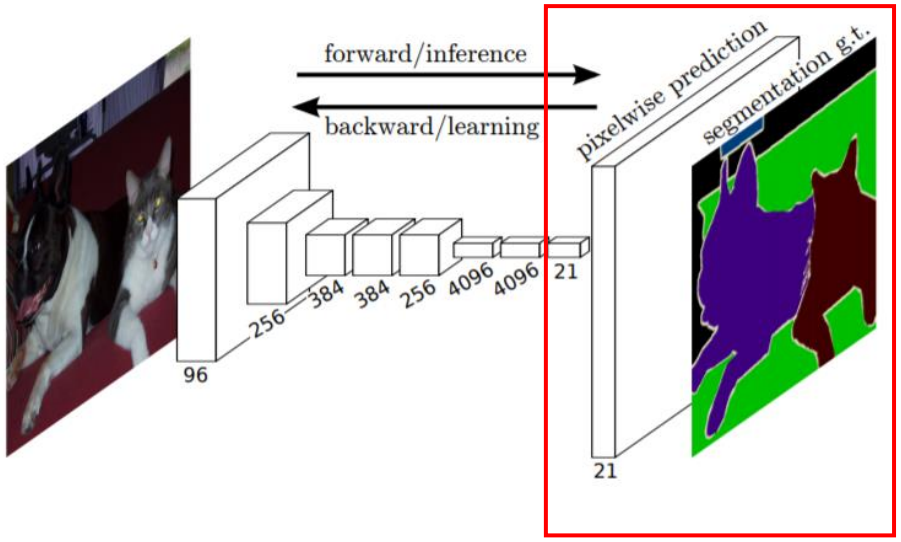
[IC-Net 제안배경]

- Baseline: PSP-Net
- 성능 저하 최소화 (약 70% mIOU)
- **real-time** 속도(30fps) 의 모델

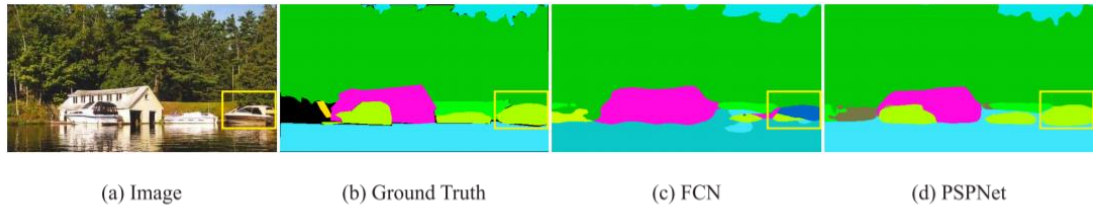
2. Pyramid Scene Parsing Network

01) Overview

[Fully Convolutional Networks (FCN)]



Problem



context 정보 부족

→ Mismatched relationship

→ Confusion categories

car
boat
water

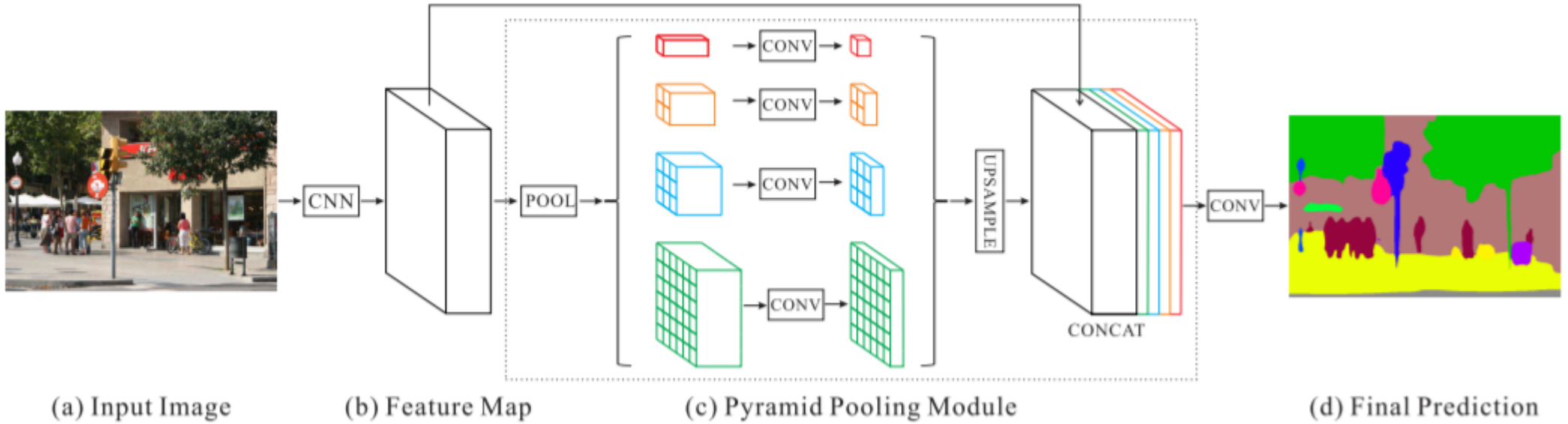
- Fully Convolutional Network(FCN)의 한계점을 해결하고자 제안된 semantic segmentation model

  → FCN 한계점: context 정보 부족으로 인한 pixel 분류성능 하락

2. Pyramid Scene Parsing Network

01) Overview

[Pyramid Scene Parsing Network (PSPNet)]



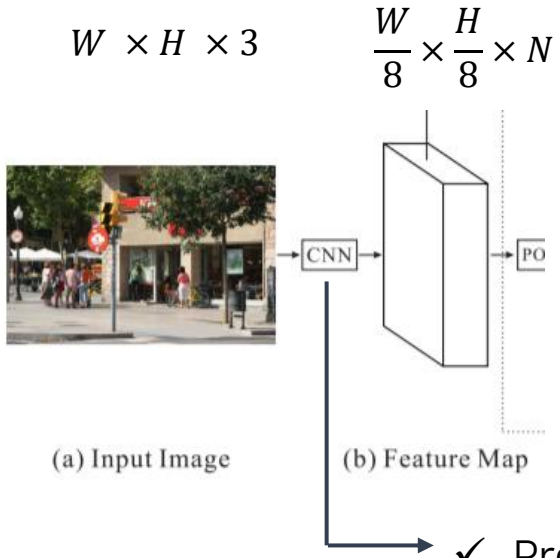
- Fully Connected Network(FCN)의 한계점을 해결하고자 제안된 semantic segmentation model

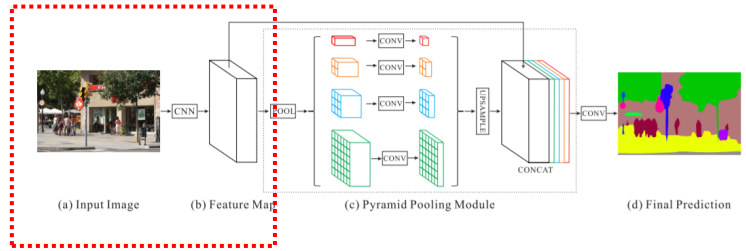
  → FCN 한계점: context 정보 부족으로 인한 pixel 분류성능 하락
- 제안 방법론: **pyramid pooling module** 추가
- 당시 ImageNet scene parsing, Cityscapes, PASCAL VOC 등 대부분 데이터에서 SOTA 성능 기록

2. Pyramid Scene Parsing Network

[1] PSP-Net (CVPR, 2017)

02) Model Architecture





✓ Pretrained CNN (with classification task)

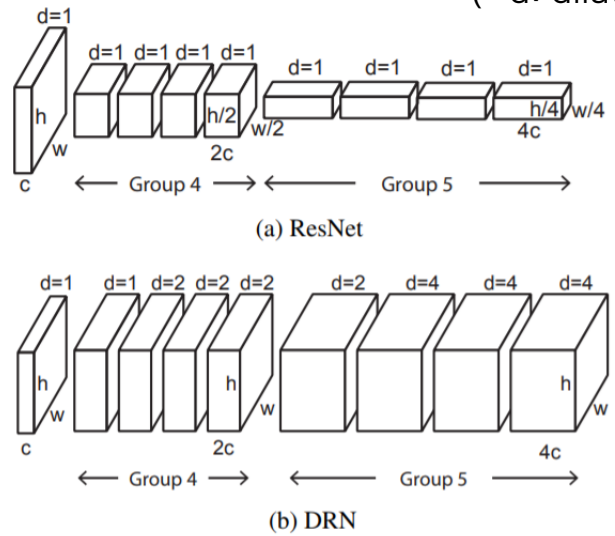
- Dilated Residual Network

(= ResNet with dilated convolution)

- Backbone: ResNet-50,**<u>101</u>**,152,269

[Dilated Residual Network (CVPR, 2017)]

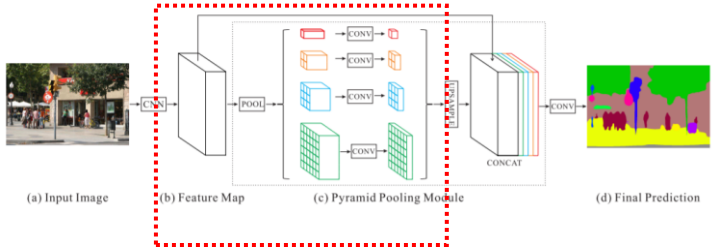
(* d: dilated rate)

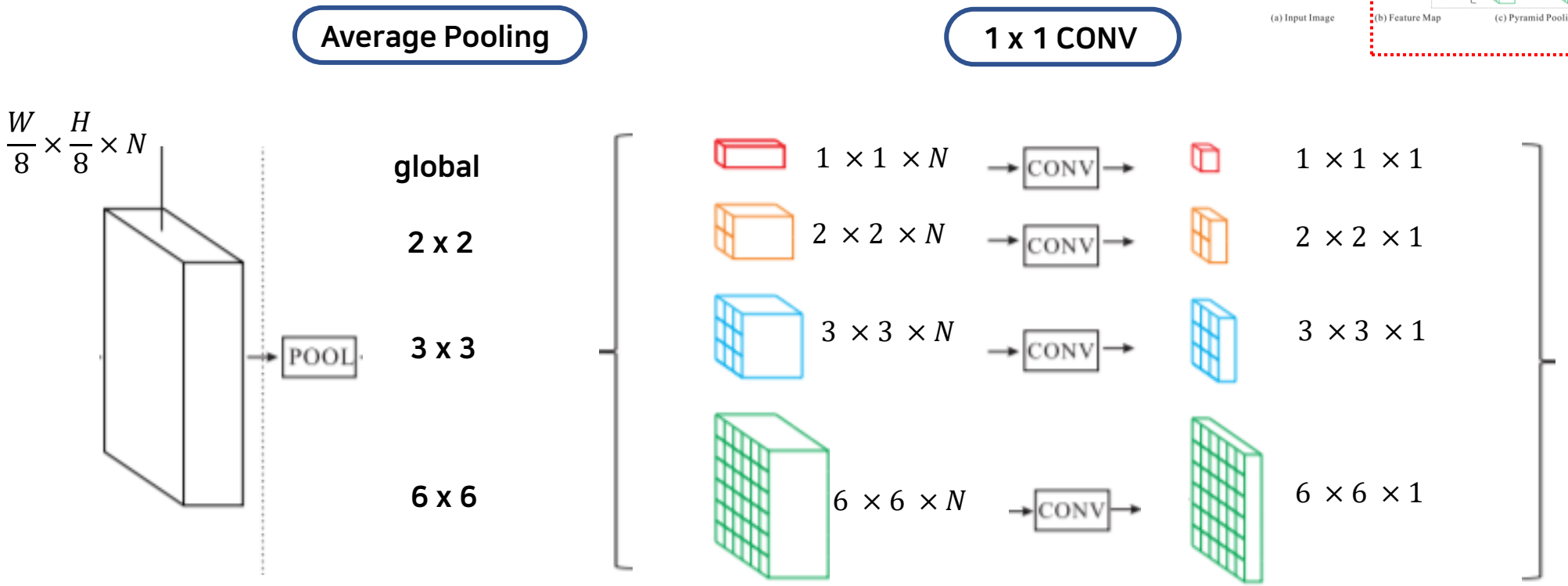


2. Pyramid Scene Parsing Network

[1] PSP-Net (CVPR, 2017)

02) Model Architecture



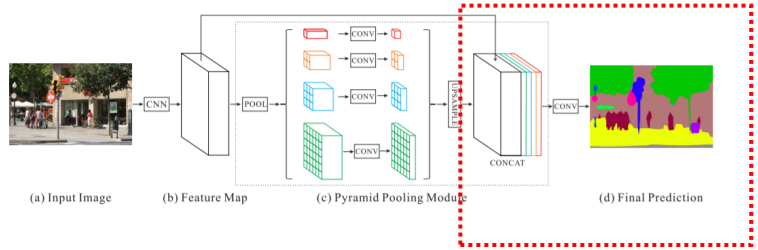


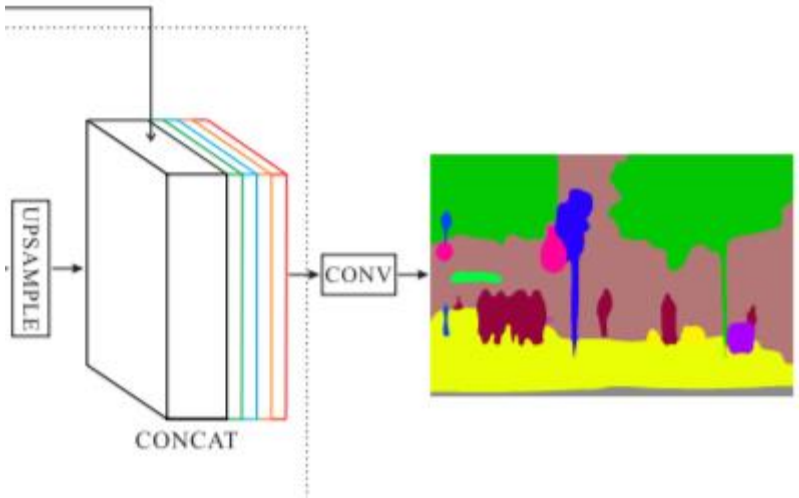
✓ Pyramid Pooling Module

- 여러 scale 의 average pooling 을 통해 rich context 를 포함한 feature map 만듦

2. Pyramid Scene Parsing Network

02) Model Architecture





✓ Upsampling

- Bilinear interpolation
- 원래 feature map 과 동일한 크기를 갖도록 함

✓ Feature Map Fusion

- 여러 단계의 feature map concatenate

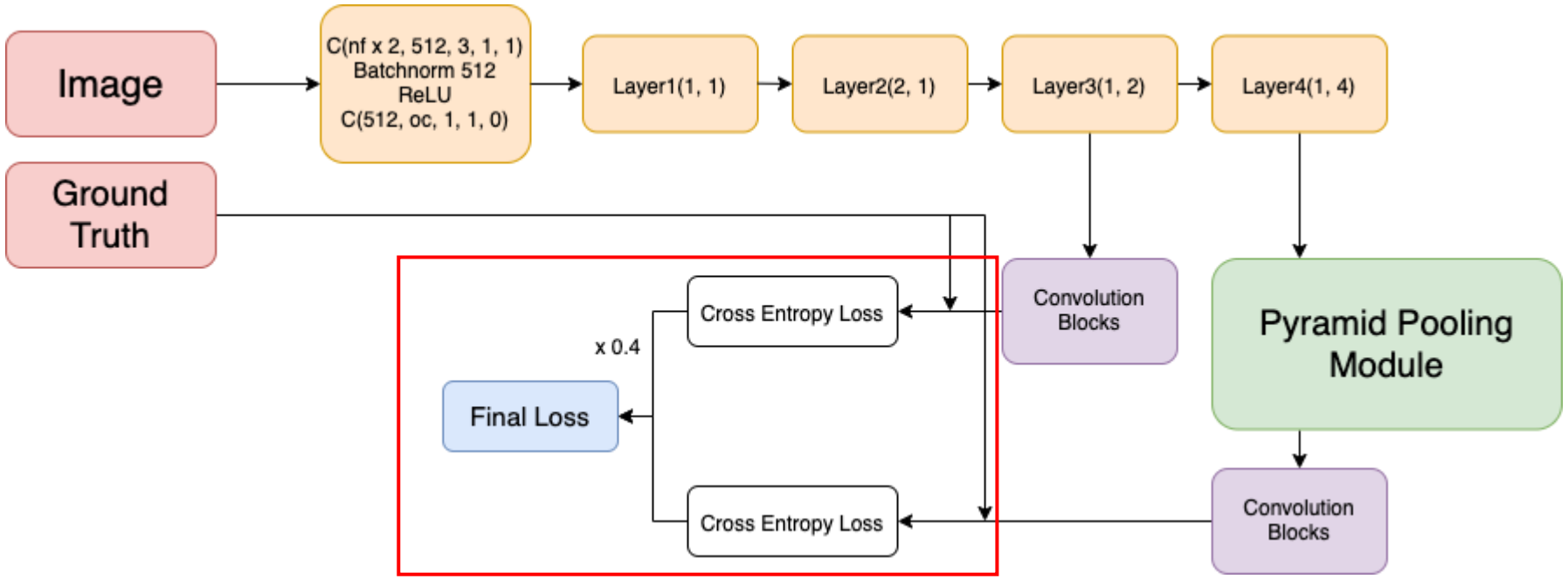
✓ Final layer

- 3x3 conv, 1x1 conv 적용
- 예측하고자 하는 class 개수를 channel 로 갖는 tensor 출력

2. Pyramid Scene Parsing Network

03) Loss Function

참고자료



- ✓ Cross entropy: pixel-wise classification 이기 때문
- ✓ Auxiliary Loss 사용:
- gradient vanishing 문제 해소 + 학습 향상
- 중간 layer 결과와 GT 간 loss 계산을 통해 0.4 만큼 최종 loss 에 반영

Summary

[1] PSP-Net (CVPR, 2017)

Pyramid Scene Parsing Network

Hengshuang Zhao[1] Jianping Shi[2] Xiaojuan Qi[1] Xiaogang Wang[1] Jiaya Jia[1]
[1]The Chinese University of Hong Kong [2]SenseTime Group Limited
{hszhao, xjqi, leojia}@cse.cuhk.edu.hk, xgwang@ee.cuhk.edu.hk, shijianping@sensetime.com

모델 특징

- Multi-scale Feature map Fusion
  (Pyramid Pooling Module)
- Auxiliary Loss

성능 측면

- mIOU, pixel accuracy 측면에서 성능 매우 좋음
- 속도는 0.78 fps 로 매우 느림

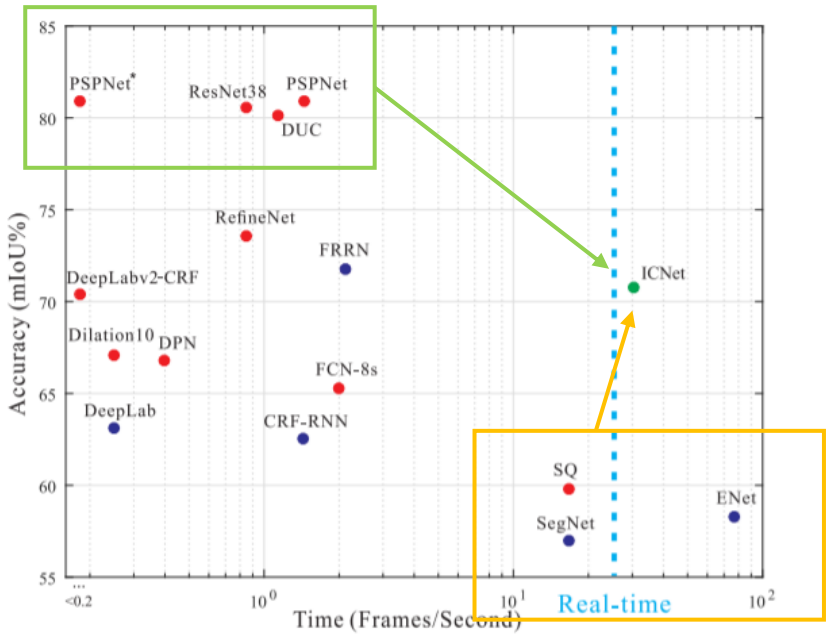
⇒ 정확도는 높지만, inference 속도가 매우 느림

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

01) Overview

성능 좋지만,
매우 느림



(a) Inference speed and mIoU

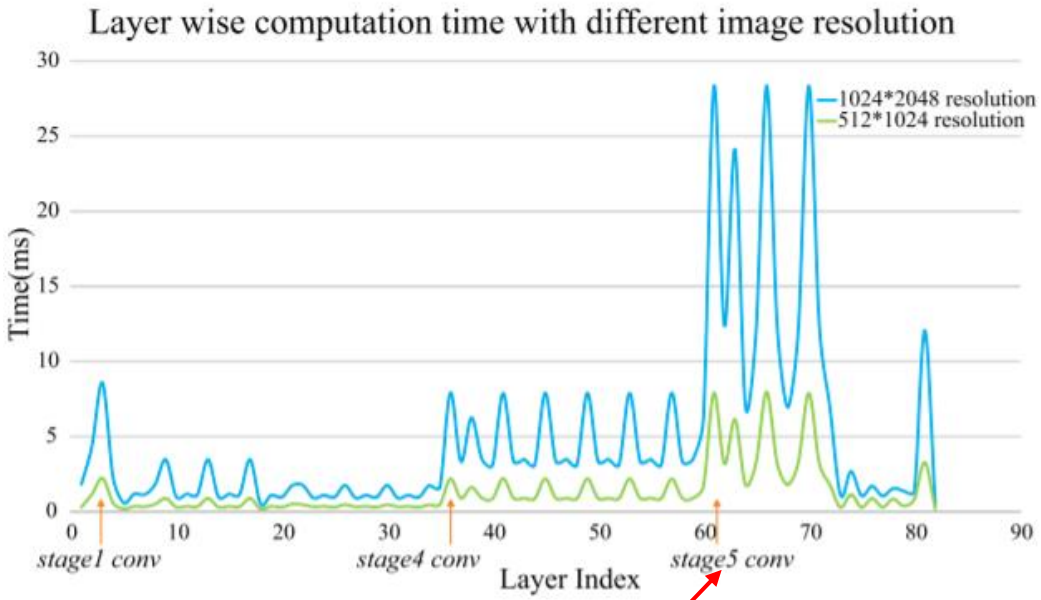
빠르지만,
성능 매우 나쁨

성능 좋은 PSP-Net 에서

속도 느리게 하는 원인 찾아 개선해보자!

3. Image Cascade Network

01) Overview



input feature map: $V \in \mathbb{R}^{c \times h \times w}$

output feature map: $U \in \mathbb{R}^{c' \times h' \times w'}$

transformation function: $\Phi : V \rightarrow U$

<u>computational cost in convolution layer</u>

$$O(\Phi) \approx c'ck^2hw/s^2$$

(c: channel, h: height, w: width, s: stride)

Problem

원인 ①: 고해상도(h, w)일수록 시간복잡도 지수 증가

원인 ②: channel 크기(c) 에 따라 시간복잡도 증가

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

01) Overview

Problem

원인 ①: 고해상도(h, w)일수록 시간복잡도 지수 증가

원인 ②: channel 크기(c) 에 따라 시간복잡도 증가

(Semantic segmentation은 dilated conv 사용하므로, 뒷단으로 갈수록 feature map 크기 증가)

⇒ **높은 이미지 해상도와 conv 연산 시 channel 증가로 시간 복잡도가 높아짐**

Solution

1) 속도 개선: 저해상도 이미지에만 heavy CNN 적용

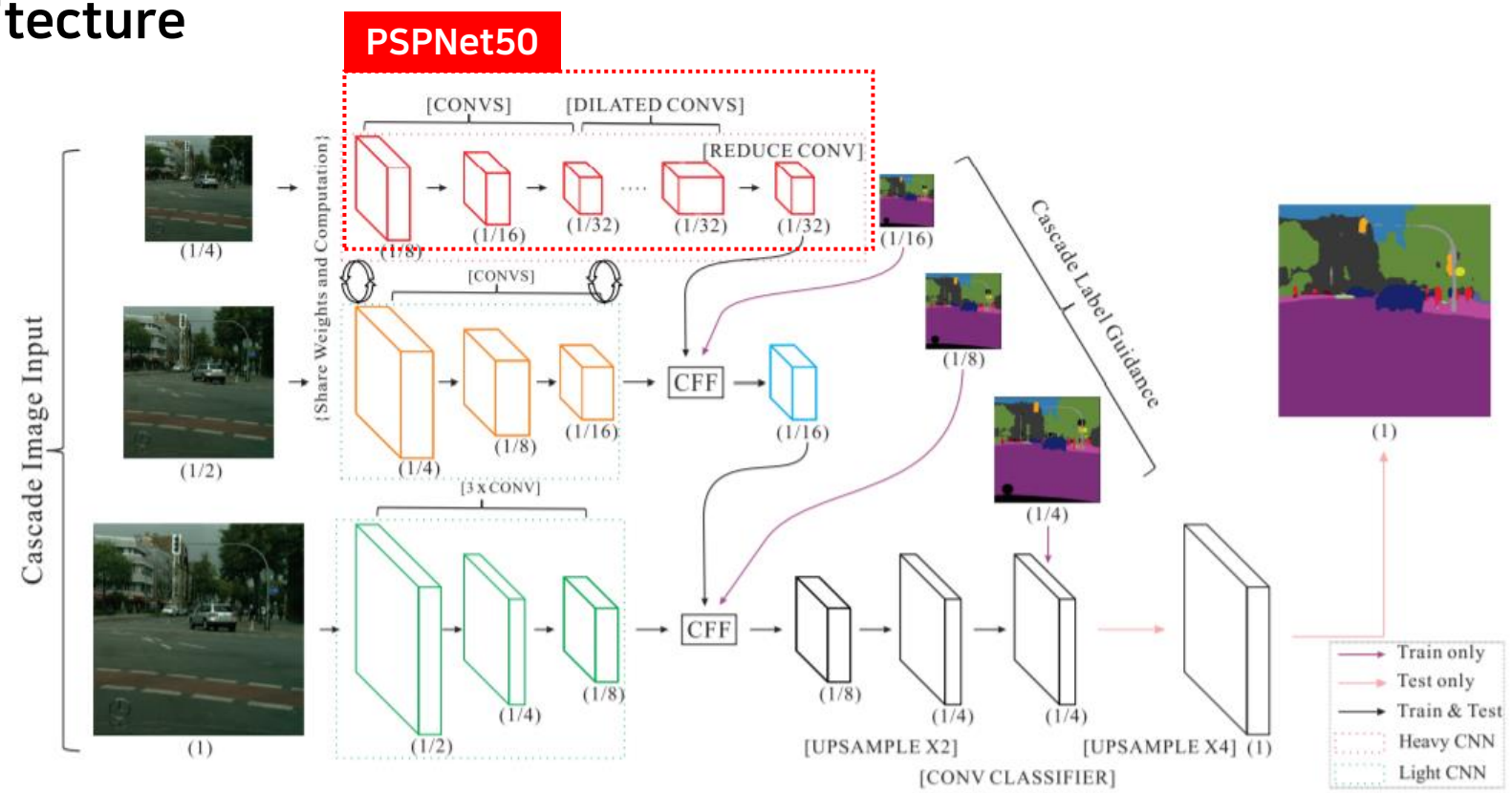
2) 성능 감소 방지:

Cascade Feature Fusion with Cascade Label Guidance

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

02) Model Architecture



- ✓ 이미지를 1, ½, ¼ 로 줄여 각 branch의 input 으로 활용
- ✓ 저해상도 이미지 (¼)는 PSP-Net50을 통해 rich semantic information 추출
- ✓ 고해상도 이미지 (½, 1)는 conv 연산을 적게(17개, 3개) 수행하고, 여기서 얻은 feature map을 저해상도 feature map 과 더해주어 coarse prediction 보강

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

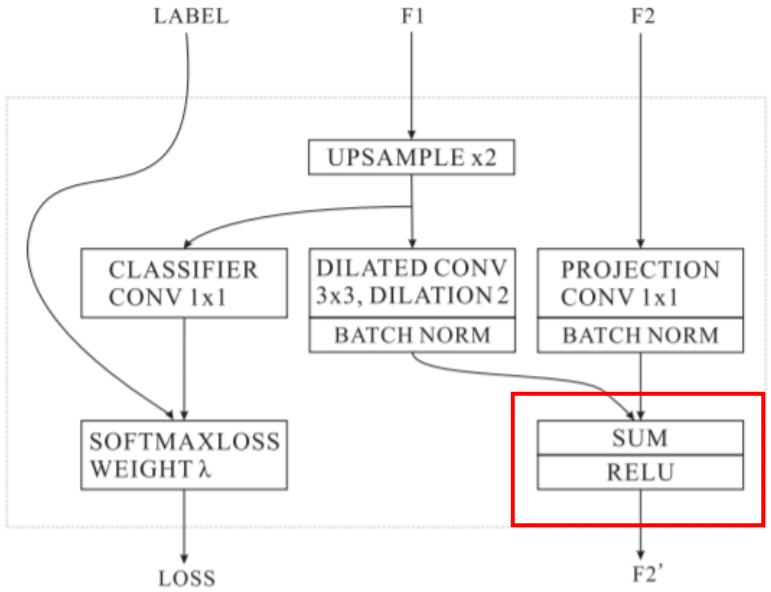
02) Model Architecture

Cascade Feature Fusion (CFF)

$F_1: C_1 \times H_1 \times W_1$ $F_2: C_2 \times H_2 \times W_2$

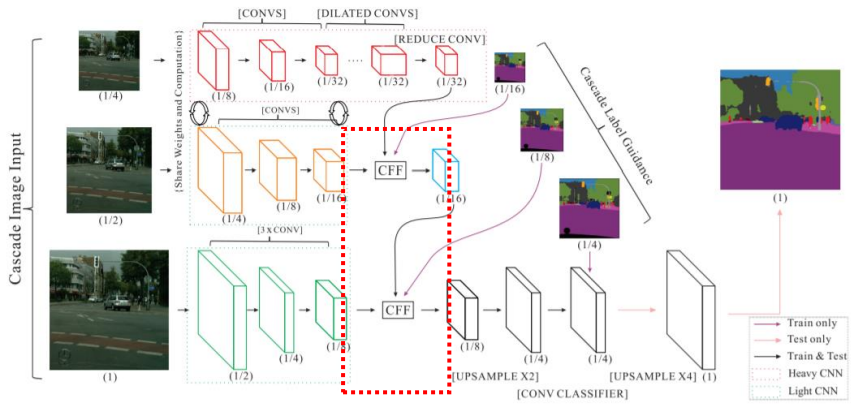
저해상도 고해상도

feature map feature map



$F_2': C_3 \times H_2 \times W_2$

Fused feature map



- CFF 사용하는 이유

rich semantic 정보 + low branch에서 놓친 boundary와 같은 세부 정보 모두 포함하기 위함

- F_1 에 bilinear interpolation 후 dilated conv 적용하는 이유

→ Kernel size 줄여서 연산시간 감소 위함

(bilinear interpolation은 연산시간 거의 걸리지 않음)

- Train & Inference 모두 적용

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

02) Model Architecture

Cascade Label Guidance

$$F_1: C_1 \times H_1 \times W_1$$

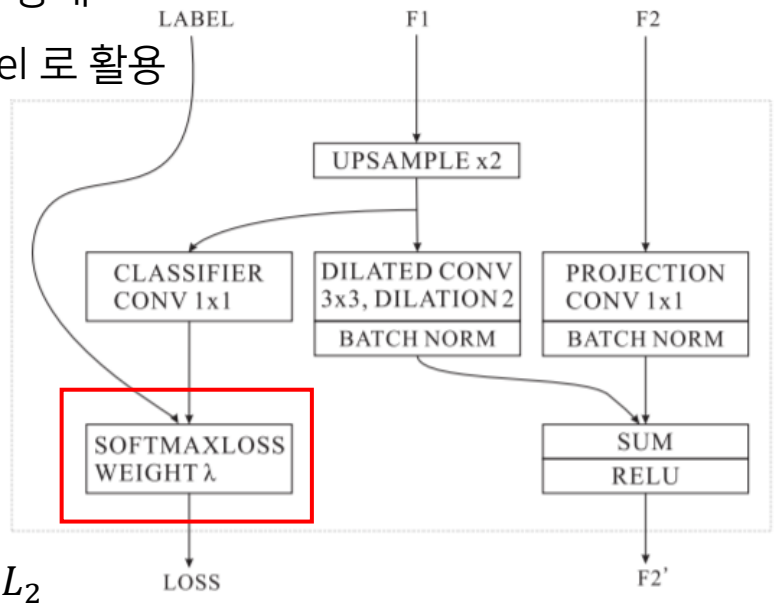
저해상도

feature map

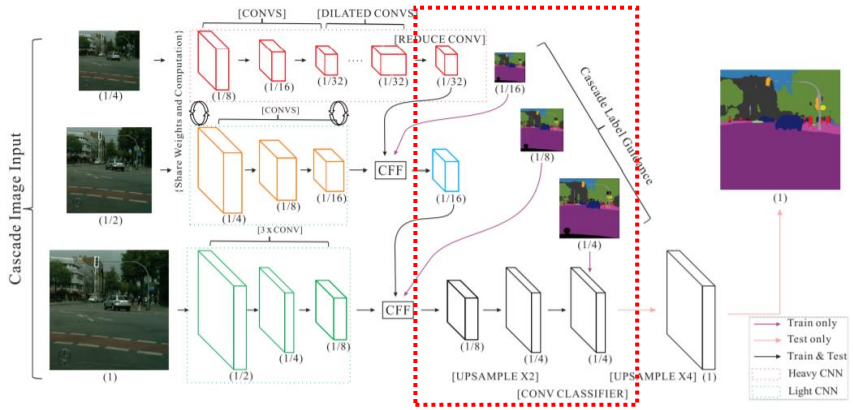
원본 label downsampling 통해

각 branch별 groundtruth label 로 활용

$$GT\ label: 1 \times H_2 \times W_2$$



$$L = \lambda_1 L_1 + \lambda_2 L_2$$



- 원본 이미지의 $^1/_{16}$, $^1/_8$, $^1/_4$ 크기의 축소된 GT label과 중간 prediction 값과 비교 후 branch 별 loss 계산

⇒ 각 branch별로 학습이 원활하도록 하기 위함

⇒ PSP-net에서의 auxiliary loss 와 유사

- Train 시에만 적용

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

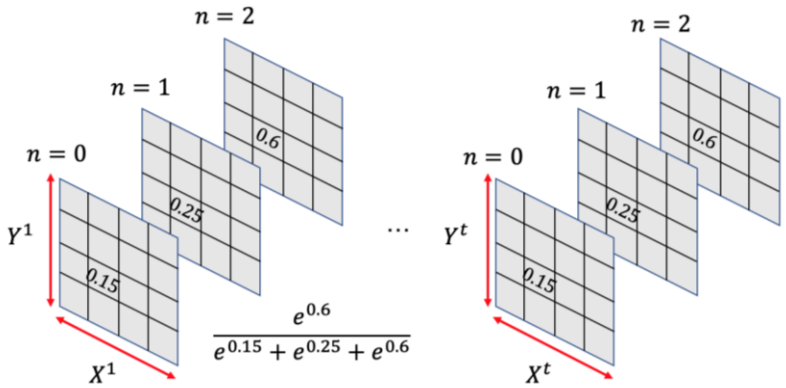
03) Loss Function

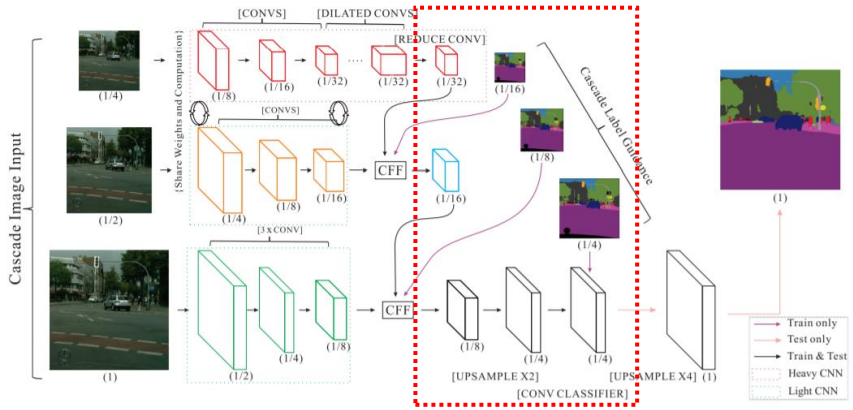
- Loss function: weighted softmax cross entropy

$$\mathcal{L} = -\sum_{t=1}^{\mathcal{T}} \lambda_t \frac{1}{\mathcal{Y}_t \mathcal{X}_t} \sum_{y=1}^{\mathcal{Y}_t} \sum_{x=1}^{\mathcal{X}_t} \log \frac{e^{\mathcal{F}^t_{\hat{n},y,x}}}{\sum_{n=1}^{\mathcal{N}} e^{\mathcal{F}^t_{n,y,x}}}$$

모든 pixel에 대한 softmax cross entropy

각 branch에서 계산한 loss 에 대해 weighted sum





$\mathcal{F}^t$: predicted feature map in branch t

$\mathcal{X}_t, \mathcal{Y}_t$: spatial size of feature map $\mathcal{F}^t$

$\mathcal{F}^t_{n,y,x}$: value at position (n, y, x) in branch t

$\hat{n}$: GT value at position (y, x)

$\mathcal{N}$: # of categories

T: # of branches $(= 3)$

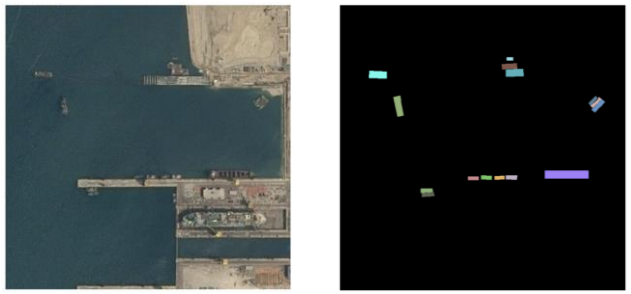
3. Image Cascade Network

04) Evaluation Metric

Accuracy

Speed

[Pixel Accuracy]



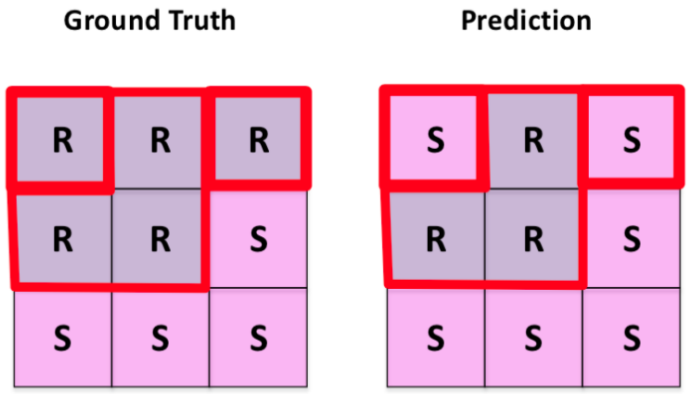
(Pixel Acc: 95% - class imbalance)

-전체 pixel 중 분류 잘한 pixel의 비율

-한계점: class imbalance

-예시 이미지 pixel acc: $^{7}/_{9}$ ($\approx 77.7\%$)

[mean IOU (mIOU)]



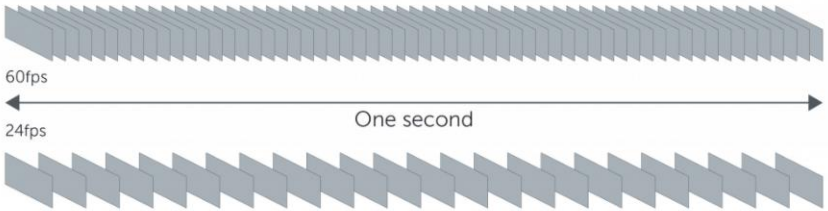
(예시: Road, Sidewalk 의 binary class)

-IOU: GT와 예측영역의 교집합/두 영역의 합집합

-mIOU: class 별 IOU 의 평균

-예시 이미지 mIOU: $(^{3}/_{5} + ^{4}/_{6}) / 2$ ($\approx 63.3\%$)

[frame per second (fps)]

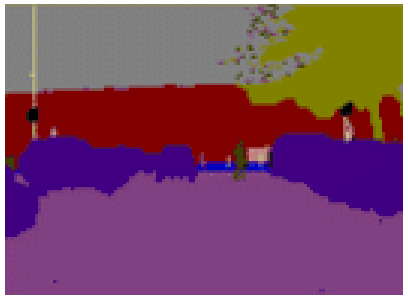
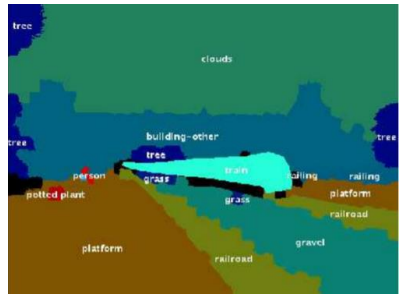


- 1초에 처리하는 frame 수
- 높을수록 빠른 속도임을 의미

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

05) Dataset

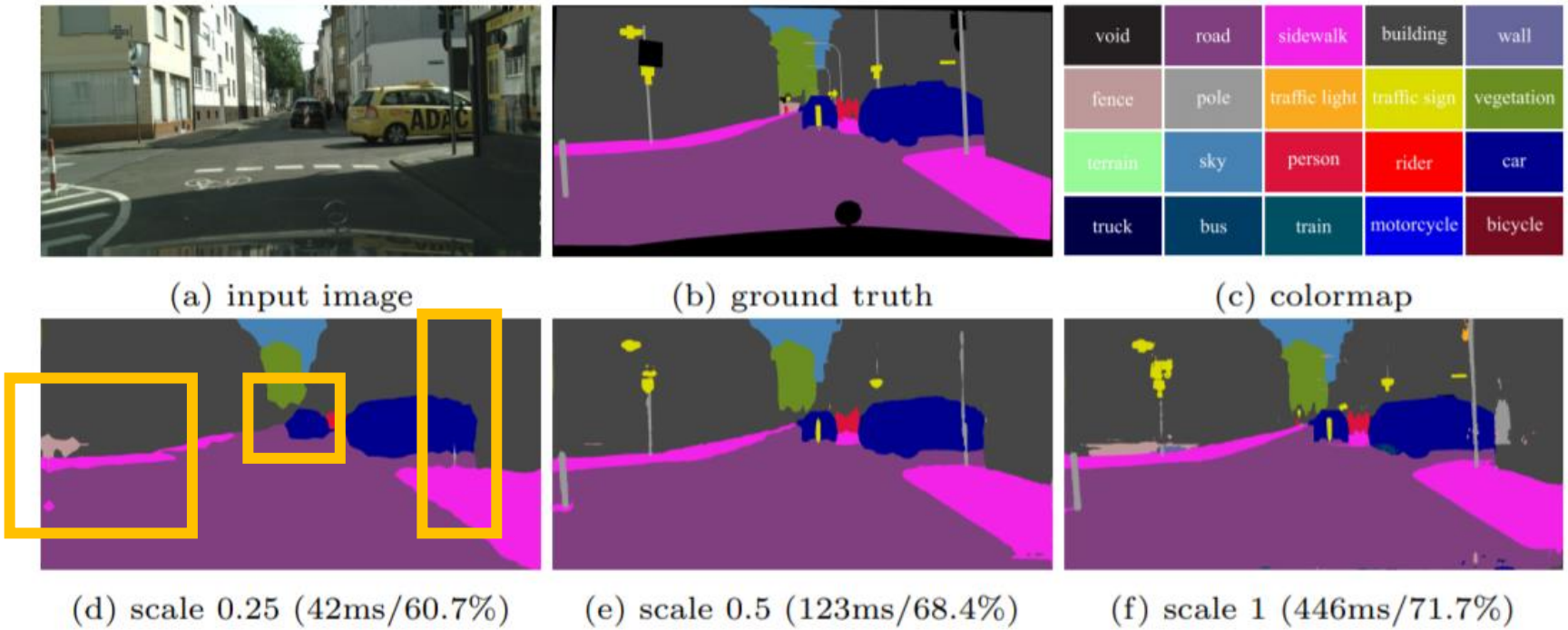
	[Cityscapes]	[CamVid]	[COCO-stuff]
Image			
Label			
	-urban scene understanding dataset -resolution: 1024 x 2048 -# images: 5000 (trn:dev:tst = 2975:500:1525) -# classes: 19	-images extracted from video sequence -resolution: 720 x 960 -# images: 700 (trn:dev:tst = 367:100:233) -# classes: 11	-labeled data based on COCO data -resolution: 640 x 640 -# images: 10K (trn:tst=9K:1K) -# classes: 182

06) Experiment: Apply 3 speedup strategy

① Downsampling input image with PSP-Net

: missing part

ms: network forward time



- Input size를 줄이는 비율이 커질수록 성능은 하락 (blurry, missing boundary)
- Input size가 줄어들수록 처리속도는 개선됨

06) Experiment: Apply 3 speedup strategy

② Downsampling feature map with PSP-Net

Downsample Size	8	16	32
mIoU (%)	71.7	70.2	67.1
Time (ms)	446	177	131

- Feature map 크기 감소에 따라 성능하락 폭 작음
- 32배 줄었을 때, inference 속도가 크게 개선됨
- 하지만 131ms 로는 10fps 에도 못 미침

③ Model Compression by reducing # of kernel

Kernel Keeping Rates	1	0.5	0.25
mIoU (%)	71.7	67.9	59.4
Time (ms)	446	170	72

- # of kernel 감소에 따라 성능하락 폭이 큼
- channel ¼ 감소해도, 여전히 inference time 느림

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

06) Experiment: Performance of IC-Net

[Performance Per branch]

Items	Baseline	sub4	sub24	sub124
mIoU (%)	67.9	59.6	66.5	**67.7**
Time (ms)	170	18	25	33
Frame (fps)	5.9	55.6	40	**30.3**
Speedup	1×	9.4×	6.8×	**5.2**×
Memory (GB)	9.2	0.6	1.1	1.6
Memory Save	1×	15.3×	8.4×	**5.8**×

* Baseline: half-compressed PSP-Net

Sub4: ¼ 축소한 이미지만 활용

Sub24: ¼ , ½ 축소한 이미지 활용

Sub124: ¼ , ½, 1 이미지 모두 활용

- Sub124 는 Baseline 대비 mIOU 거의 유사하며 속도도 5배 빠름

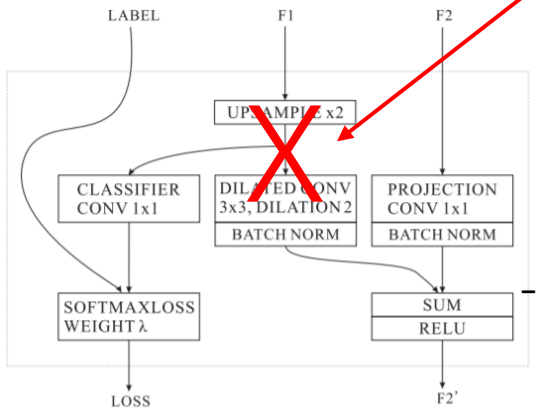
[Effectiveness of CFF & Cascade Label Guidance]

DC3	DC5	DC7	CFF	CLG	mIoU (%)	Time (ms)
✓				✓	66.7	31
	✓			✓	66.7	34
		✓		✓	68.0	38
			✓	✓	67.7	33
			✓		66.8	33

DC3: deconvolution (kernel: 3x3)

CFF: Cascade Feature Fusion

CLG: Cascade Label Guidance



- CFF & CLG 를 사용할 때 Time 대비 mIOU 좋음

3. Image Cascade Network

[2] IC-Net (ECCV, 2018)

06) Experiment

[Cityscapes]

Method	DR	mIoU (%)	Time (ms)	Frame (fps)
SegNet [3]	4	57.0	60	16.7
ENet [8]	2	58.3	13	76.9
SQ [9]	no	59.8	60	16.7
CRF-RNN [16]	2	62.5	700	1.4
DeepLab [2]	2	63.1	4000	0.25
FCN-8S [1]	no	65.3	500	2
Dilation10 [14]	no	67.1	4000	0.25
FRRN [12]	2	71.8	469	2.1
PSPNet^3 [5]	no	81.2	1288	0.78
ICNet	no	69.5	33	30.3

* DR: 기존 해상도에서 감소시킨 비율

[CamVid]

Method	mIoU (%)	Time (ms)	Frame fps
SegNet [3]	46.4	217	4.6
DPN [15]	60.1	830	1.2
DeepLab [2]	61.6	203	4.9
Dilation8 [14]	65.3	227	4.4
PSPNet50 [5]	69.1	185	5.4
ICNet	67.1	36	27.8

[COCO-stuff]

Method	mIoU (%)	Time (ms)	Frame fps
FCN [1]	22.7	169	5.9
DeepLab [2]	26.9	124	8.1
PSPNet50 [5]	32.6	151	6.6
ICNet	29.1	28	35.7

- ✓ IC-Net 이 모든 dataset 에 대해 fps 대비 mIOU 가장 좋음
- ✓ 고해상도 cityscapes 데이터에 대해 30fps 로 real-time 수준의 inference 속도 보임

Summary

[2] IC-Net (ECCV, 2018)

ICNet for Real-Time Semantic Segmentation on High-Resolution Images

Hengshuang Zhao[1], Xiaojuan Qi[1], Xiaoyong Shen[2], Jianping Shi[3], Jiaya Jia[1,2]

[1]The Chinese University of Hong Kong, [2] Tencent Youtu Lab, [3]SenseTime Research
{hszhao,xjqi,leojia}@cse.cuhk.edu.hk,
dylanshen@tencent.com, shijianping@sensetime.com

모델 특징

- Cascade Feature Fusion
- Cascade Label Guidance
- Fast with High resolution image

성능 측면

- [ICNet] 69.5% mIOU, 33 fps
- [PSPNet] 81.2% mIOU, 0.78 fps

⇒ 정확도가 적게 감소, inference 속도가 매우 빠름

Personal Research

개인연구

1) 연구주제: 라즈베리 파이와 segmentation 모델을 활용한 Real time Lane Detection

2) 진행상황

- 강의 수강: [라즈베리파이] IoT 딥러닝 computer vision (수강완료)
- 라즈베리 파이와 MNIST 분류모델 통한 Real time 인식
- Semantic segmentation 활용한 lane detection 모델과 코드 공부

- 참고논문:
    - Learning Lightweight Land Detection CNNs by Self Attention Distillation (ICCV, 2019)

- 모바일 디바이스에서도 실행가능한 E-Net 의 encoder에 self attention 을 적용하여 성능을 높인 Enet-SAD 모델 제안함.
- 이미지 내에서 주목해야 하는 부분에 attention score 를 높게 줌.

개인연구

3) 발생 문제 및 해결

[발생문제1] torch 버전 호환 문제

- 로컬(1.7)과 라즈베리(1.4)에서의 torch 버전 불일치

  → 로컬에서 모델 학습 후, 라즈베리파이에서 inference 불가

- 해결방법: 로컬 환경을 라즈베리와 일치시킨 후, 모델 학습

```
# Convolutional neural network (two convolutional layers)
class ConvNet(nn.Module):
    def __init__(self, num_classes=10):
        super(ConvNet, self).__init__()
        self.layer1 = nn.Sequential(
            nn.Conv2d(1, 16, kernel_size=7, stride=2, padding=1),
            nn.BatchNorm2d(16),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer2 = nn.Sequential(
            nn.Conv2d(16, 32, kernel_size=5, stride=1, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.fc = nn.Linear(7*7*32, num_classes)
```

[발생문제2] CNN 분류기의 낮은 성능 차이 문제 해결

- 원인: CNN 적용 시, 첫 번째 conv layer 의 filter size를 작게 주어 이미지에 대한 receptive field 가 좁아 이미지의 특징을 잘 반영한 feature map 이 생성되지 않았음.

- 해결방법: 기존에는 3x3 filter 로 하였는데, 이를 5x5, 7x7 로 확대하니 성능이 급격히 향상됨.

⇒ 첫 번째 layer 에서는 일반적으로 kernel size 를 크게 (5x5, 3x3) 으로 주어서 비교적 큰 receptive field 로 정보 얻음.

감사합니다