

딥러닝 4단계

W2L08 – W3L01

Seung-mok Lee

Sunkee kim's lab, Seoul Nat'l Univ.

COSINE Collaboration

Sep. 22nd, 2020

Contents

ConvNets 을 위한 실질적인 조언들	
	오픈소스를 활용해라 Update : 2020.07.13 ♥ 5
	전이학습 Update : 2020.07.13 ♥ 6
	데이터 확대 Update : 2020.07.13 ♥ 5
	컴퓨터 비전 시스템을 구축하기 위한 팁 Update : 2020.07.13 ♥ 4
탐지 알고리즘	
	물체 로컬리제이션 Update : 2020.07.14 ♥ 7

Open Source

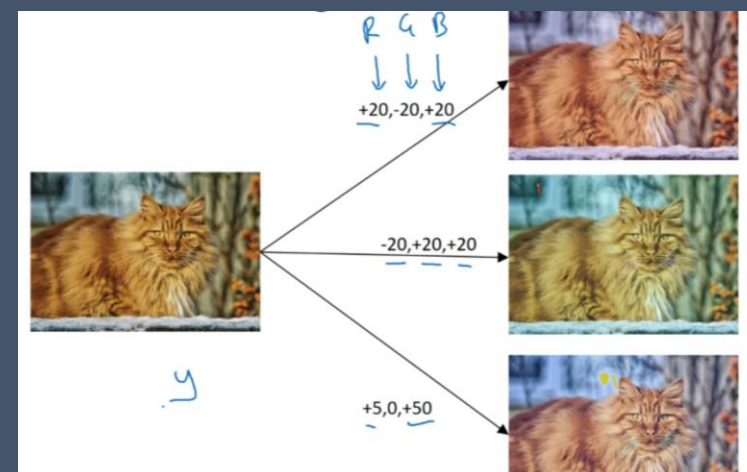
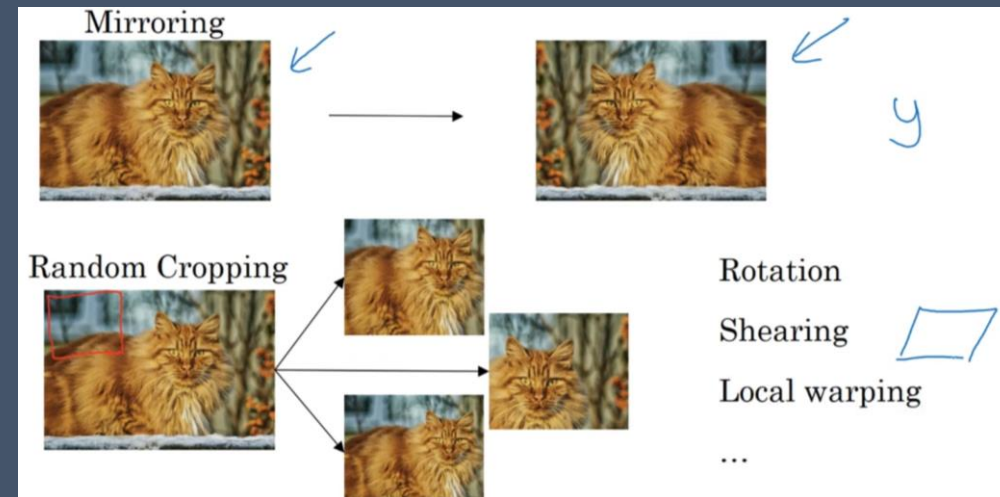
- Github에 검색하면 좋은 네트워크들이 이미 구현되어 있다.
- 웬만하면 그냥 찾아서 써라!
 - 박사급 학생들도 논문만 보고 구현하는 것은 뻥세다.
- Git 사용법에 익숙해진다면 아주 좋을 것이다.

Transfer Training

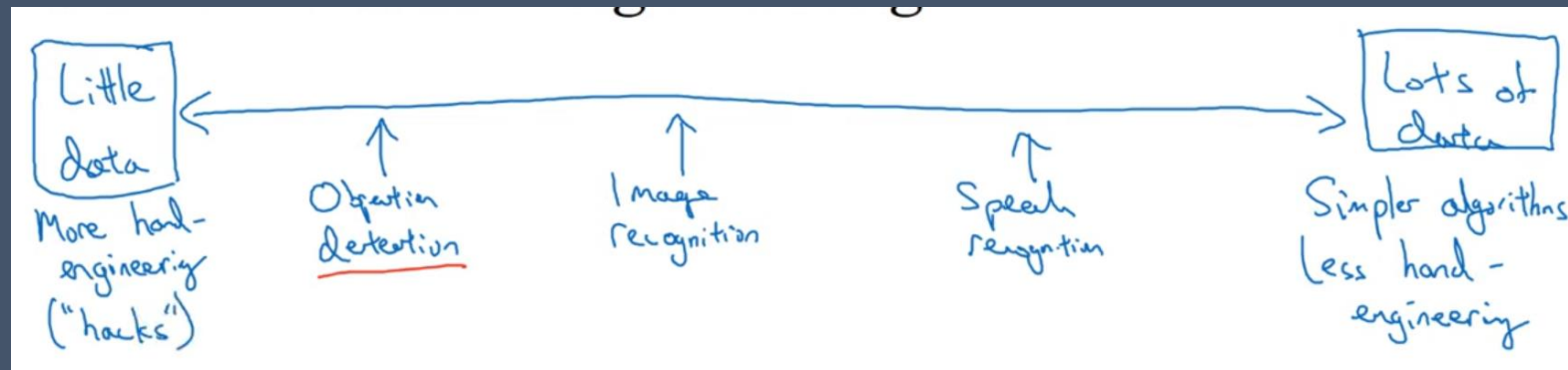
- 특히 컴퓨터 비전 앱을 만들고 있다면, 누군가가 이미 학습시킨 네트워크를 다운받아서 전이학습 시켜라.
 - 앞의 레이어들은 학습이 불가능하게끔 freeze 시키고 마지막 층만 다시 학습시킨다.
 - 원한다면, 그리고 데이터가 충분하다면 뒤의 몇 층을 다시 학습시켜도 괜찮다.
 - 데이터가 아주 충분하다면 아예 처음부터 학습시켜도 좋다. 하지만 보통은 인터넷에 올라온 네트워크들이 더 큰 데이터에 더 잘 학습되었을 것이다.

Data Augmentation

- Vertical/horizontal mirroring
- Random cropping
- Rotation, Shearing, Local warping, ...
- Color shifting



Spectrum of DL Problem



- 데이터가 많으면 아무렇게나 만들고 학습시키면 된다.
- 데이터가 적으면 Hand-engineering이 필요하다.
 - Data에 잘 맞는 구조 (CNN)
 - 직접 특성 뽑아내기

Competition and Benchmarks

- 효율적인 알고리즘 개발을 독려하는 좋은 현상!
- 하지만 일부 Competition에만 유용하고 실제로는 잘 사용되지 않는 알고리즘도 있다.
 - 서비스하기에는 속도가 현저히 느려지거나 메모리가 너무 많이 소요된다.

Ensembling 3-15 networks → $\hat{y}$

- Train several networks independently and average their outputs

Multi-crop at test time

개인적으로는 이 방법들을 실제 제품에는 절대 사용하지 않습니다

- Run classifier on multiple versions of test images and average results

10-crop

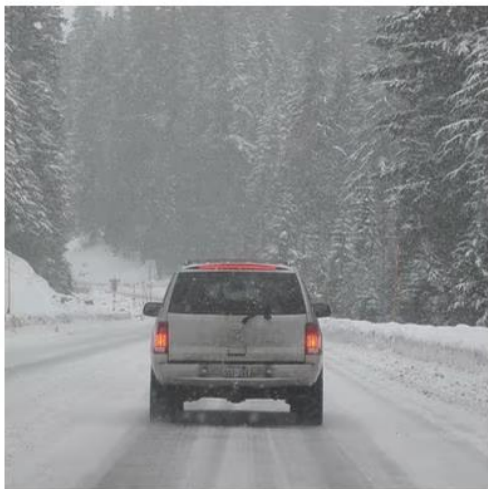


VIDEOS

1 + 4 + 1 + 4

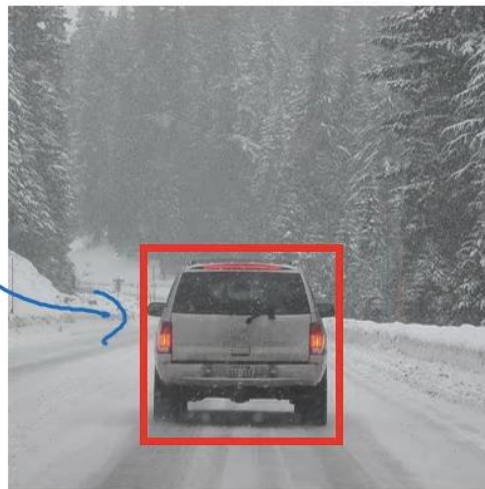
Object Localization

Image classification



"Car"

Classification with localization

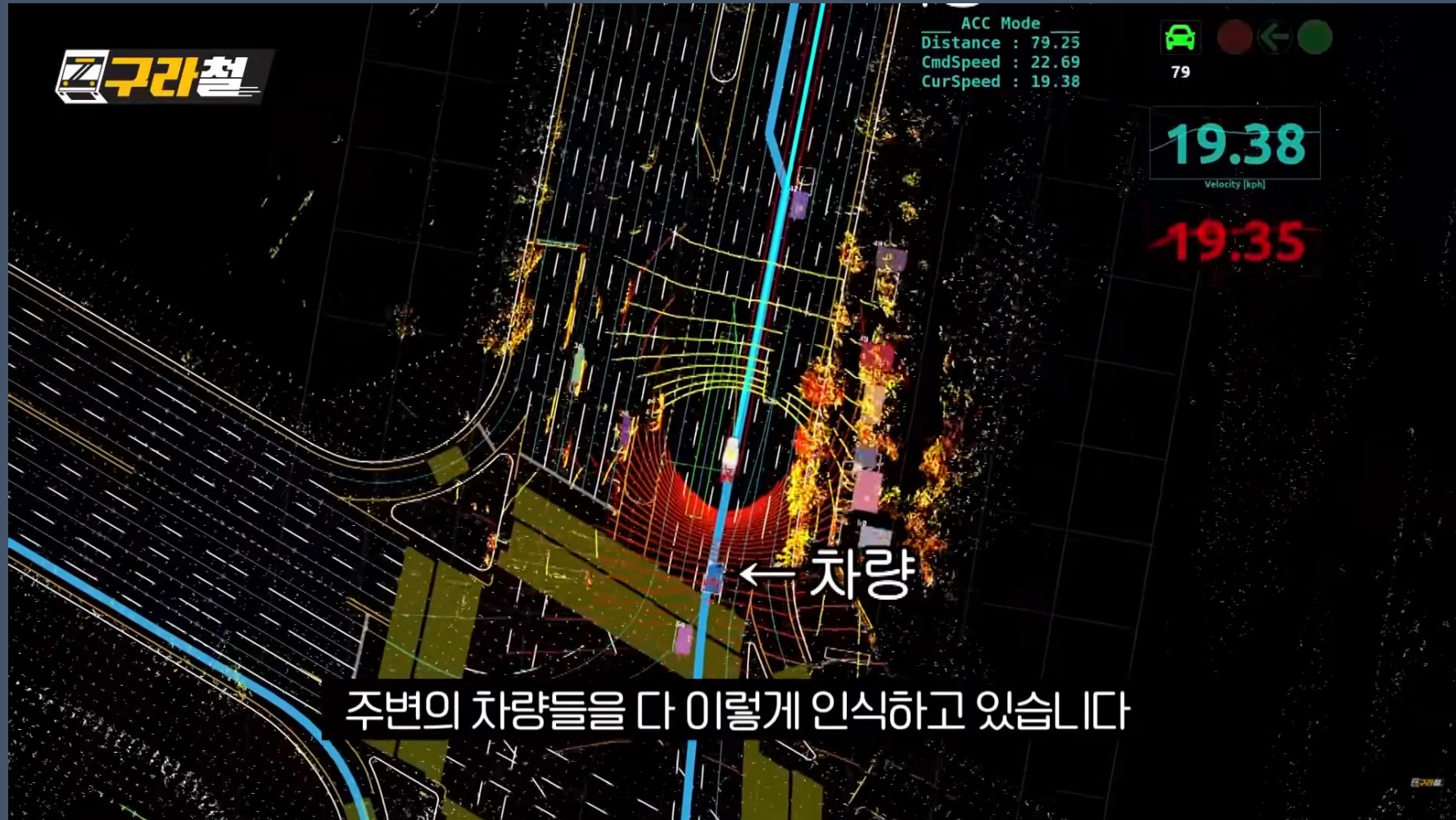


"Car"

Detection

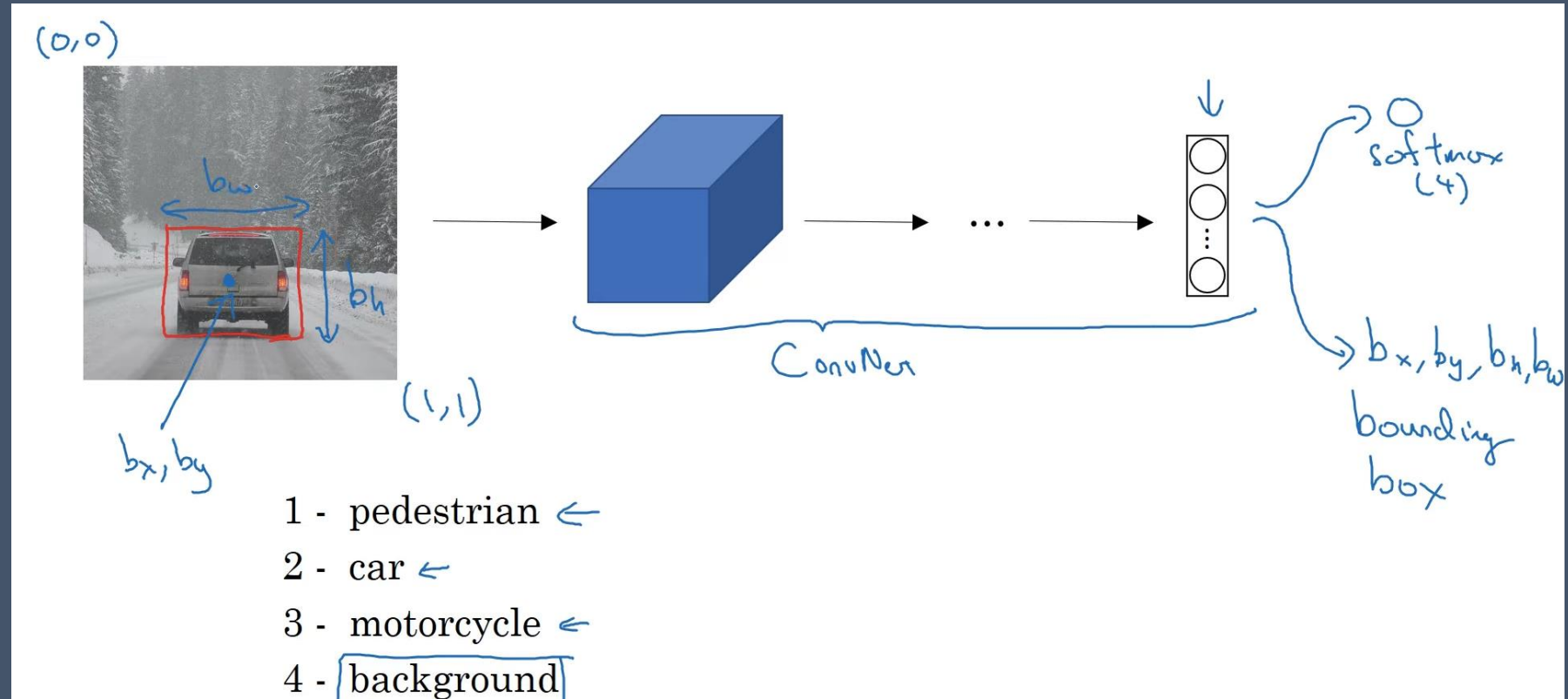


Object Detection Example



<https://www.youtube.com/watch?v=GAFnPRyC6ng&t=549s>

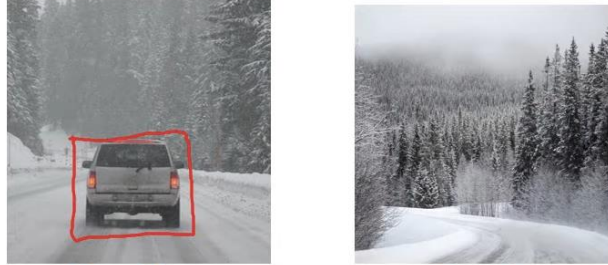
Object Detection Network Structure



Object Detection Output & Loss

1 - pedestrian
2 - car ←
3 - motorcycle
4 - background ←

Need to output b_x, b_y, b_h, b_w , class label (1-4)

$X =$ 

$Y =$ $\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$ is there any object?

$L(\hat{y}, y) = \begin{cases} (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \dots + (\hat{y}_8 - y_8)^2 & \text{if } y_1 = 1 \\ (\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 \end{cases}$

$\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$

$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$ ← "don't care"

- Loss function에서 c_1, c_2, c_3 에 대해서는 cross-entropy를, p_c 에 대해서는 logistic regression을 사용하기도 한다.

LAST SLIDE