

# Zero shot Object Detection

2018, ECCV, A. Bansal et al.

한양대학교  
컴퓨터 소프트웨어 학과  
인공지능 연구실  
조건희

# Introduction

---

- Zero Shot Detection

- Zero Shot Learning + Object Detection
- detect object class which are not observed during training

- Zero Shot Learning 개요

- 제로샷은 이미지 학습과 더불어 이미지 외의 부가 정보(text embedding 등)를 활용하여 학습하지 않은 클래스를 분류하기 위한 classification task
- 많은 CNN 기반의 이미지 학습은 annotated 데이터가 충분히 있어야 학습이 수월함.
- 만약 분류하려는 클래스의 수가 적은 경우에는 데이터를 확보하기 용이하지만, 클래스 수가 아주 많은 경우에는 기존 방법만으로는 학습하기에 충분한 데이터를 확보하기 어려움.
- 제로샷 러닝은 이런 문제를 해결하기 위한 접근방식.

# Related work

## Related work

---

- 일반적인 이미지 분류 (Image classification)



→ 개



→ 고양이

train

---

test



→ 개

## Related work

---

- 제로샷 이미지 분류 (Image classification)



→ 개



→ 고양이

train

---

test



→ ?

- 왜 귀찮게 제로샷 러닝을 하려고 하는가?

We have labeled data, why bother?

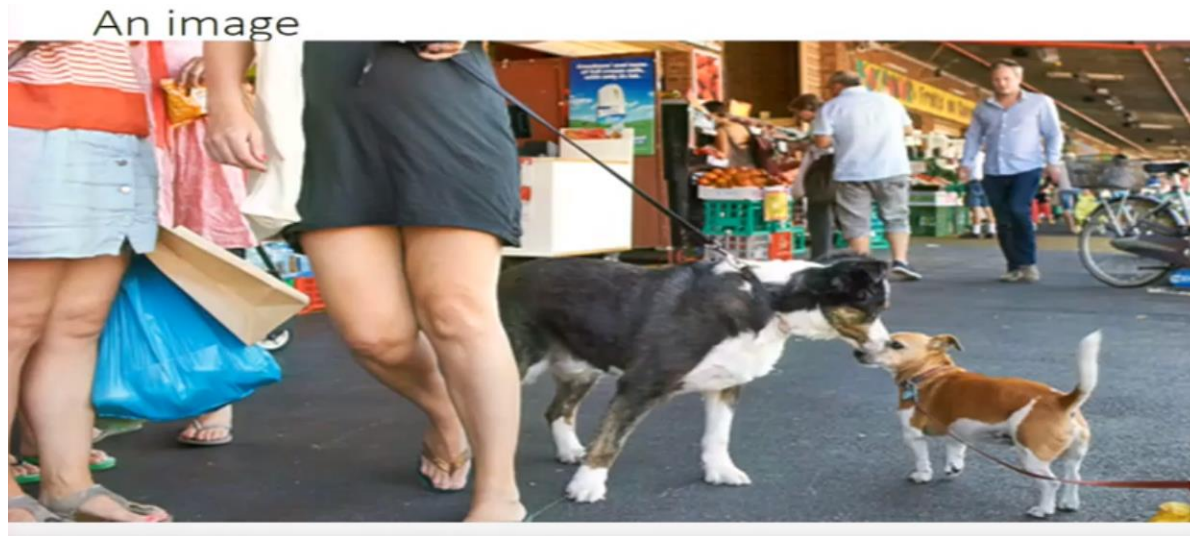


**Imagenet: ~15,000,000 images**  
**Open Images: ~9,000,000 images**  
**Places: ~2,500,000 images**

# Related work

참조 - 영상 강의(2) <https://youtu.be/dE4nU5OaQqA>

- 왜 귀찮게 제로샷 러닝을 하려고 하는가?



Classification



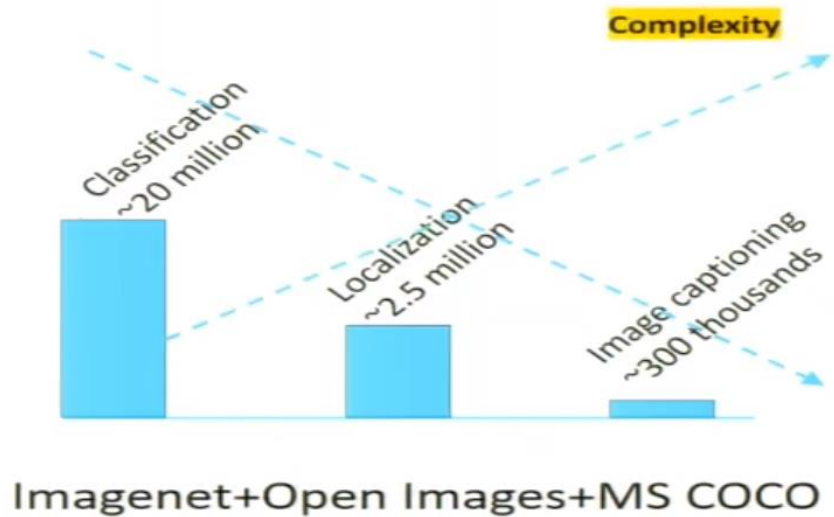
Segmentation



Captioning



- 왜 귀찮게 제로샷 러닝을 하려고 하는가?



더 복잡한 annotation 일수록 데이터셋을 구축하는 것이 더 어려움.

The more complex task we target,  
the fewer annotations we have,  
the more relevant zero-shot learning is.

## Related work

- 제로샷 이미지 분류 (Image classification)



→ 개  $\in$  *seen*



→ 고양이  $\in$  *seen*

train

test



→ 개  $\in$  *seen*



→ 말  $\in$  *unseen*

knowledge transfer with side-information

## ▪ Attributes as side-information

### ▪ AWA 데이터셋 (Animals with Attributes) [Lampert et al. CVPR'09]

▪ class : 50

▪ attribute : 85

#### otter

black: yes  
white: no  
brown: yes  
stripes: no  
water: yes  
eats fish: yes



#### polar bear

black: no  
white: yes  
brown: no  
stripes: no  
water: yes  
eats fish: yes



#### zebra

black: yes  
white: yes  
brown: no  
stripes: yes  
water: no  
eats fish: no



# Related work

참조 - <http://www.vision.caltech.edu/visipedia/CUB-200-2011.html>

- Attributes as side-information

- CUB 데이터셋 (Caltech UCSD-Birds) [Wah et al. '11]

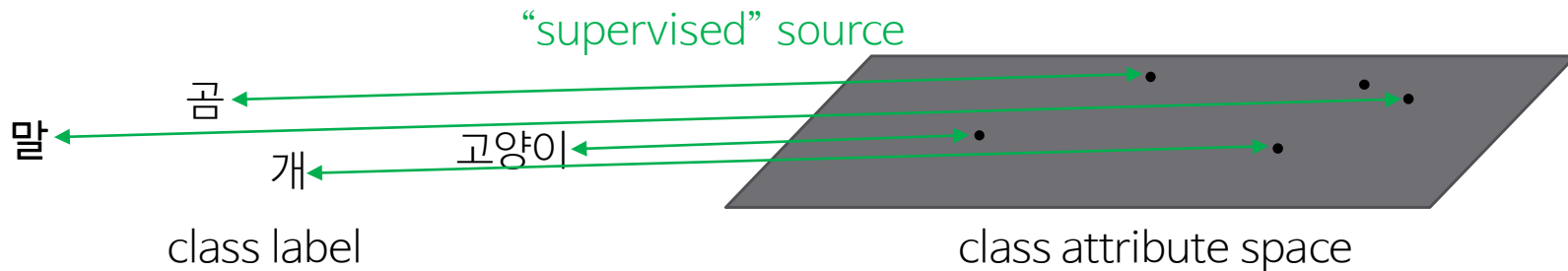
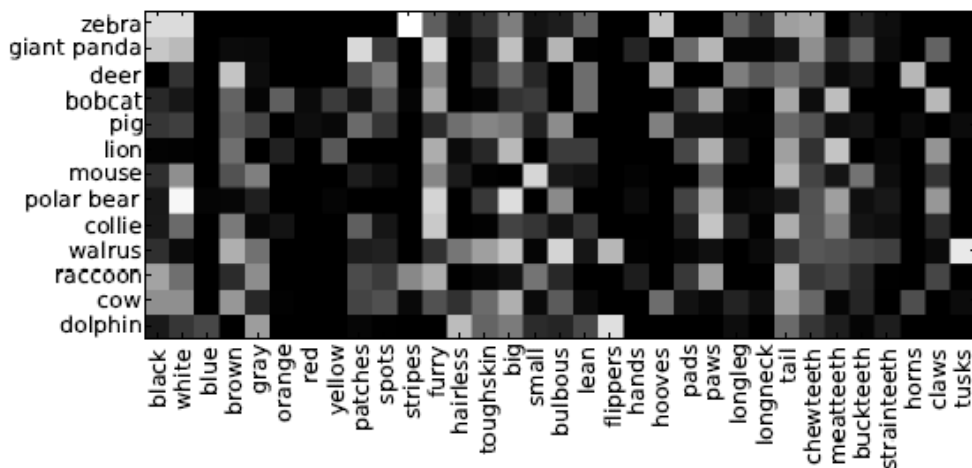
- class : 200
    - attribute : 312



## Related work

- Attributes as side-information
  - Image → Attribute 모델을 학습

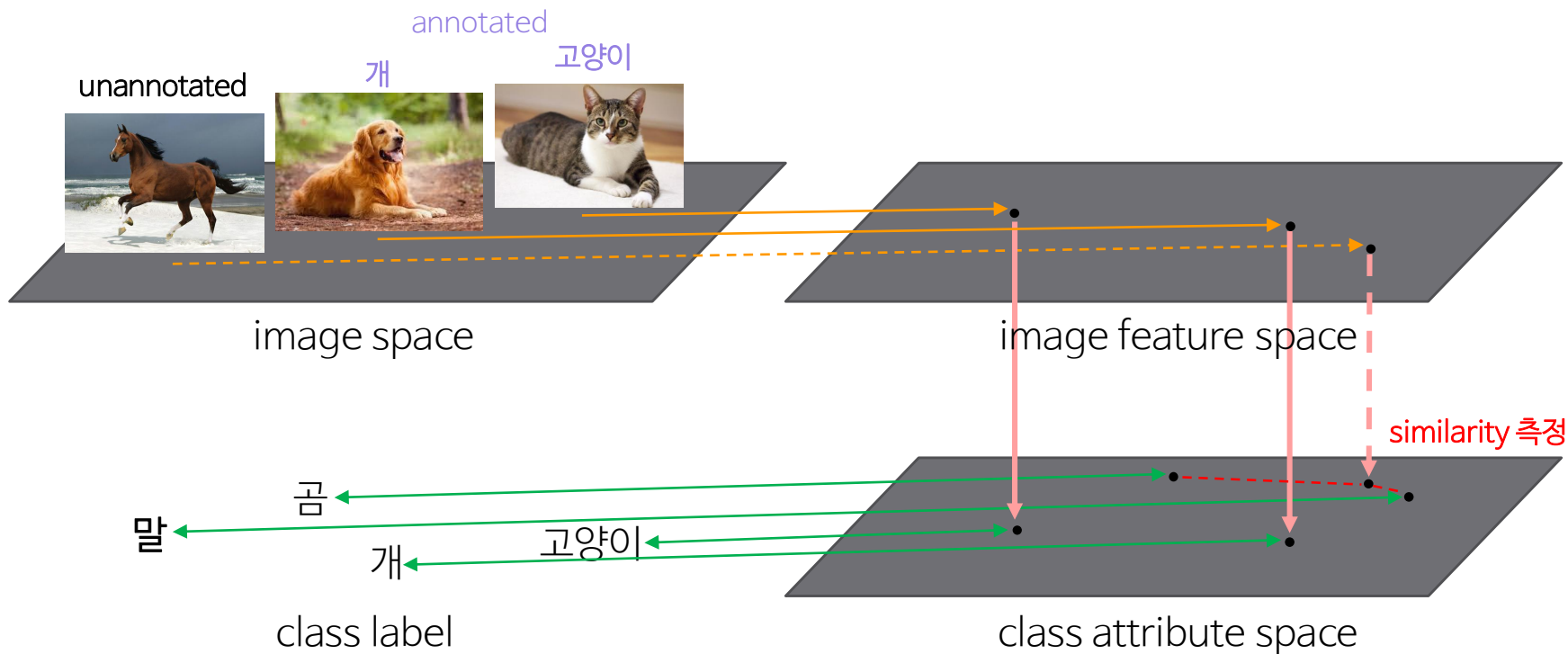
class attribute matrix



## Related work

- Attributes as side-information

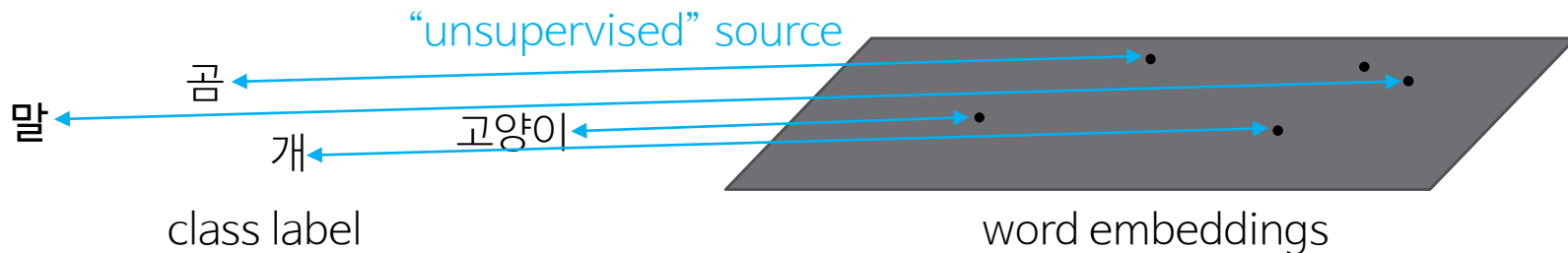
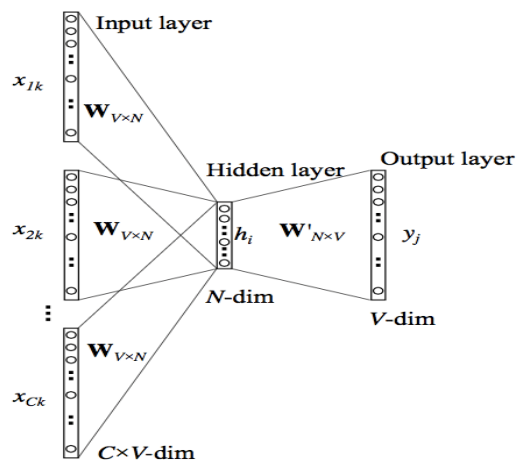
- Image → Attribute 모델을 학습
- 보통 unannotated 를 **unseen** 이라 하고 annotated를 **seen** 이라 함.



## Related work

- Semantic word-embeddings as side-information
  - Image  $\rightarrow$  word-embedding 모델을 학습

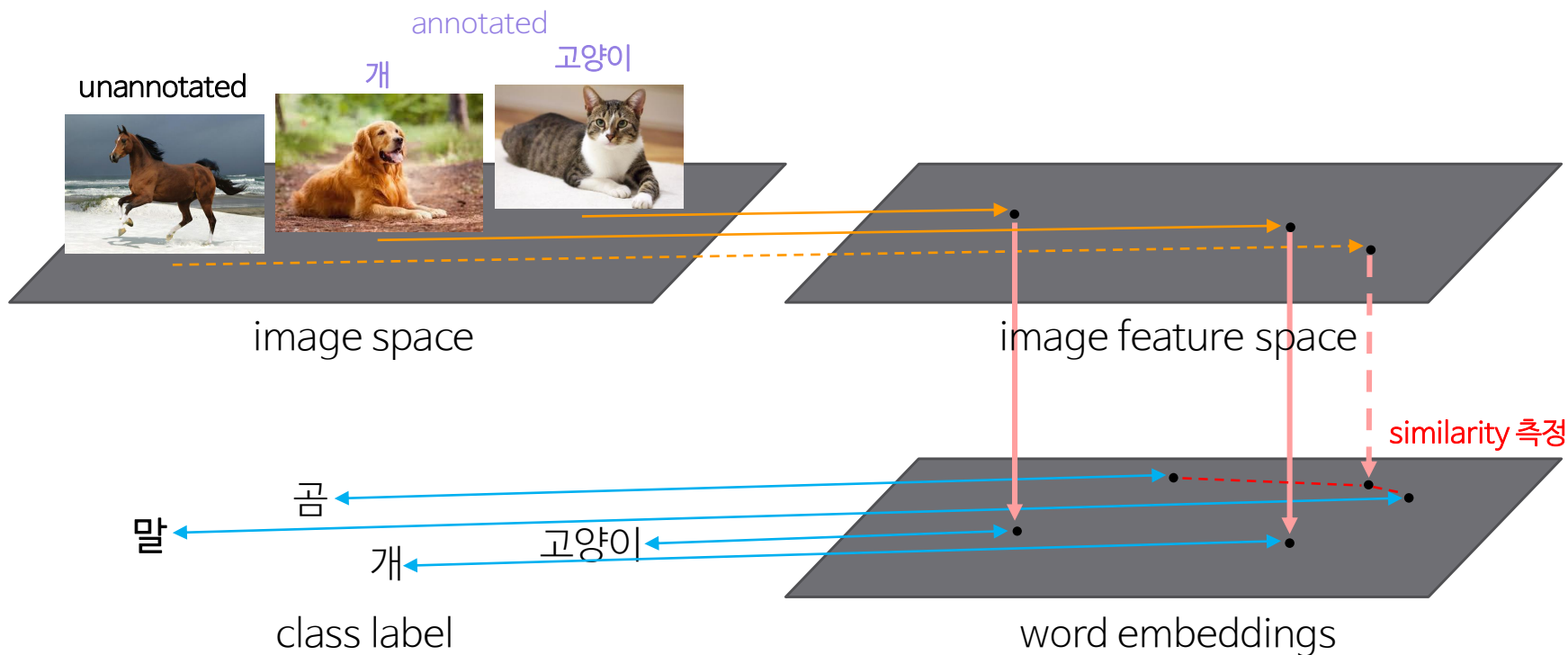
word2vec





## Related work

- Semantic word-embeddings as side-information
  - Image → word-embedding 모델을 학습





# Related work

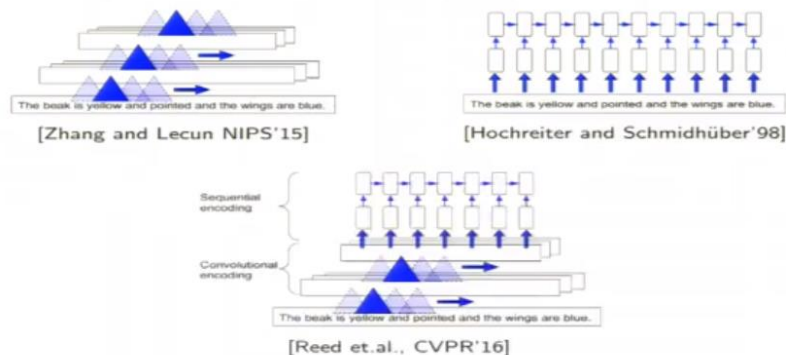
참조 - 영상 강의(2) <https://youtu.be/dE4nU5OaQqA>

## Other side-information

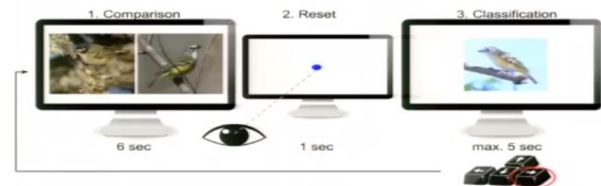
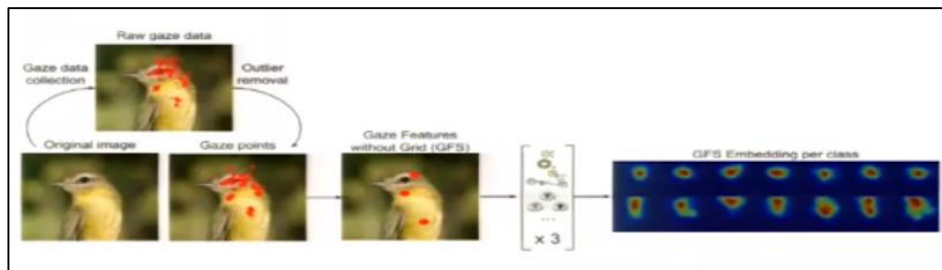
### Detailed visual description



[Reed et.al. CVPR'16, ICML'16, NIPS'16]



### Human gaze

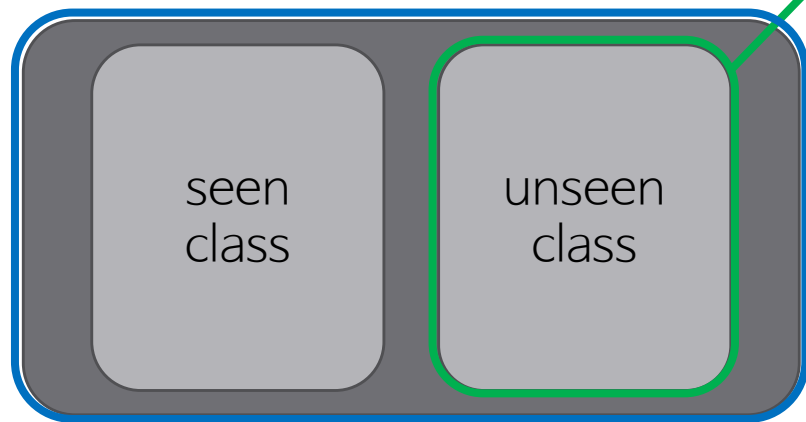


gaze features



### ▪ Classic setting VS. Generalized setting

- Classic : unseen 데이터셋에 포함된 클래스 중에서만 분류하는 것
  - 현실에서는 어떤 이미지가 주어졌을 때, 그게 학습에 참여한 클래스인지 아닌지 알려주지 않음.
  - 그래서 실용적이지 못함.
- Generalized : seen/unseen 데이터셋에 포함된 모든 클래스 중에서 분류하는 것.
  - classic 방법에 비해 실용적이지만 성능이 떨어짐.
  - state-of-the-art 성능 약 20%



여기 중에서 무슨 클래스인지 맞추면 됨.  
(Classic ZSL)

전체 중에서 무슨 클래스인지 맞춰야 함.  
Generalized ZSL

- Summary

- 왜 제로샷 러닝을 하려고 하는가?

- task 가 복잡해질 수록 annotated data 가 부족해짐.

- Side-information

- attributes
    - word embeddings
    - others(visual description, human gaze 등)

- Classic setting VS. Generalized setting

- unseen class 중 어떤건지? VS. seen class 와 unseen class 전부 중 어떤건지?

# Approach

# Approach

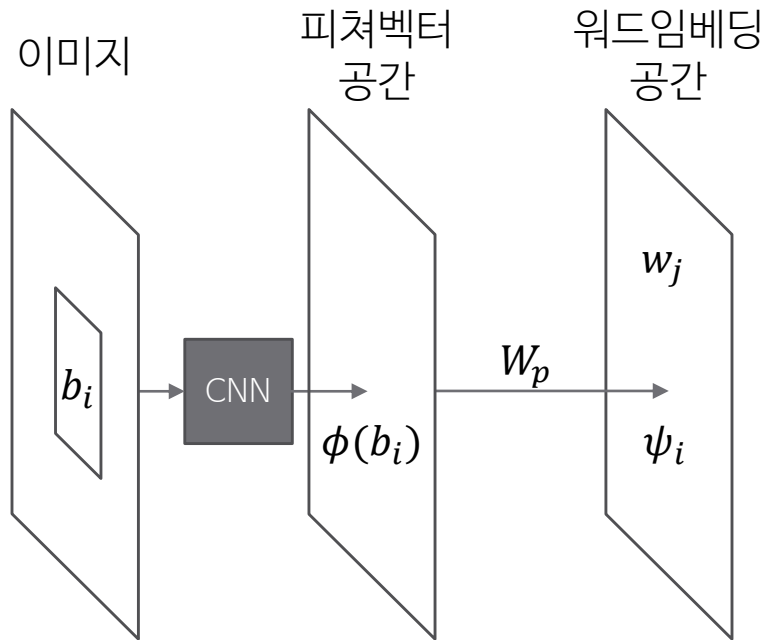
## ▪ Baseline Zero-shot Detection(ZSD)

### ▪ 모든 클래스 $\mathcal{C} = \mathcal{S} \cup \mathcal{U} \cup \mathcal{O}$

- $\mathcal{S}$  : seen class
- $\mathcal{U}$  : unseen class
- $\mathcal{O}$  : seen, unseen 에 속하지 않는 클래스

### ▪ 트레이닝

- 이미지 라벨  $y_i \in \mathcal{S}$
- 이미지 바운딩 박스  $b_i$
- semantic embedding for class label  $w \in \mathbb{R}^D$
- feature vector  $\phi(b_i)$
- linear projection matrix  $W_p$
- $\psi_i = W_p \phi(b_i)$ : 이미지 피쳐 벡터 스페이스에서 워드 임베딩 벡터 스페이스로 프로젝션한 벡터
- cosine similarity  $S_{ij}$  :  $\psi_i$ 와  $w_j$  두 벡터 사이의 코사인 유사도



# Approach

- Baseline Zero-shot Detection(ZSD)

- max-margin loss

$$L(b_i, y_i, \theta) = \sum_{j \in S, j \neq i} \max(0, m - S_{ii} + S_{ij})$$

- $S_{ii}$  : 클래스  $i$  의 워드 임베딩 벡터( $w_i$ )와 이미지로부터 프로젝션한 벡터( $\psi_i$ )와의 코사인 유사도
    - $S_{ij}$  : 클래스  $j$  의 워드 임베딩 벡터( $w_j$ )와 이미지로부터 프로젝션한 벡터( $\psi_i$ )와의 코사인 유사도
    - $m$  : margin
    - $S_{ii}$  와  $S_{ij}$ 의 차이가 margin  $m$  에 가까워지도록 CNN 과 projection matrix  $W_p$  의 파라미터를 학습
  - additional reconstruction loss [Kodirov et al. CVPR'17]

$$\min_W \|X - W^T S\|^2 + \lambda \|WX - S\|^2$$

- $X$  : feature vector ( $=\phi(b_i)$ ) ,  $W$  : projection matrix( $= W_p$ ) ,  $S$  : semantic 임베딩 공간의 클래스 라벨 워드 임베딩 벡터
    - 프로젝션 매트릭스( $W$ )로 만든 semantic 임베딩 벡터( $WX = W_p \phi(b_i)$ )가 원하는 해당하는 워드 벡터( $S$ )와 가까워지면서도 워드 벡터( $S$ )와 프로젝션 매트릭스( $W$ )로 원래의 피쳐 벡터( $X$ )를 reconstruct 하도록 학습

## ▪ Baseline Zero-shot Detection(ZSD)

### ▪ 테스트

- 먼저 바운딩 박스를 찾기 위해 Edge-Boxes [Zitnick et al. ECCV'14]의 region proposal 알고리즘 그대로 사용.
- Edge-Boxes의 proposal score가 0.07 이상인 바운딩 박스만을 오브젝트 바운딩 박스로 인식.
- 찾은 바운딩 박스 이미지를 pre-trained Inception-ResNet v2 모델에 넣어 feature vector 를 뽑고, 트레이닝한 프로젝션 매트릭스로 워드 임베딩 스페이스 상의 벡터( $\psi_i$ )를 구함.
- $\psi_i$  벡터와 unseen 클래스 라벨 임베딩 벡터( $w_j$ )와 코사인 유사도  $S_{ij}$  를 구함.

$$\hat{y}_i = \arg \max_{j \in U} S_{ij}$$

- $\hat{y}_i$  : predict 라벨
- $S_{ij}$  : 클래스  $j$  의 워드 임베딩 벡터( $w_j$ )와 이미지로부터 프로젝션한 벡터( $\psi_i$ )와의 코사인 유사도
- $j \in U$  이므로 unseen 클래스 라벨의 워드 임베딩 벡터 중에서 이미지로부터 프로젝션한 벡터와 가장 유사한 벡터를 찾아 그 라벨을 이미지의 라벨로 선택(classic setting).
- 하나의 unseen 오브젝트에 대하여 여러 바운딩 박스가 겹쳐 있을 수 있으므로, 스코어가 높은 바운딩 박스와 IoU가 0.4 이상인 바운딩 박스는 제외.

# Approach

## ▪ Baseline Zero-shot Detection(ZSD)

### ▪ background-aware의 문제

- 기존 오브젝트 디텍션 모델들은 성능 향상을 위해 학습시 'background' 클래스를 같이 학습함.
- 이렇게 학습한 후 바운딩 박스 proposal 중 아무런 오브젝트가 없는(또는 원하는 오브젝트가 없는) 바운딩 박스를 제외하기에 용이하기 때문.
- 이 논문에서는 이러한 모델을 background-aware detector 라고 함.
- 그러나 Zero-shot detection의 경우 background 의 선택은 non-trivial problem.
  - 왜냐면, 트레이닝 클래스가 아닌 클래스(unseen class)가 포함된 background 바운딩 박스가 존재할 수 있기 때문.
  - 트레이닝 시 'background' 로 학습했기 때문에 테스트 시 찾아야 할 바운딩 박스가 제외되어 버리는 문제가 생길 수 있음.
  - 예를 들어, 아래 그림에서 노란색 바운딩 박스들은 'background' 의 바운딩 박스로 학습되었을 가능성이 있음.
  - 따라서 baseline ZSD는 background 클래스를 아예 학습에서 제외함.

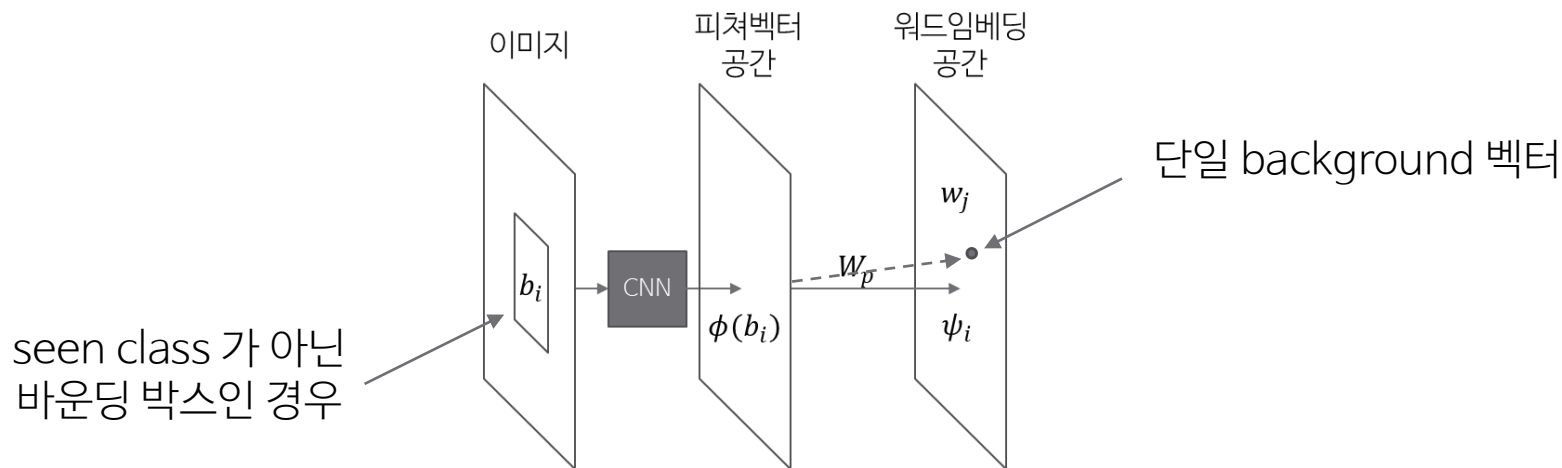




# Approach

## ▪ Background-aware zero-shot detection

- 이러한 문제로 인해 ZSD에서는 학습 시 background 의 바운딩박스 선택이 쉽지 않음.
- 이 논문에서는 이를 해결하기 위한 2가지 방법을 제시
  - (1) Statically assigned background (SB)
    - single fixed background 워드 임베딩 벡터를 사용.
    - seen class에 속하지 않는 바운딩 박스는 전부 단일 background 임베딩 벡터로 프로젝션 되도록 학습.
    - background vector는 다른 클래스들의 워드 임베딩 벡터와 비슷한 크기를 가지도록 임의로 벡터를 생성하여 사용.



# Approach

- Background-aware zero-shot detection

- 이 논문에서는 이를 해결하기 위한 2가지 방법을 제시

- (2) Latent assignment based(LAB)

- Expectation Maximization(EM) -like algorithm

- 목적 : multiple latent class를 통해 background 의 다양한 시각정보를 구분하여 학습.

- seen class 와 unseen class에 포함되지 않는 클래스 들을 large vocabulary 로부터 가져옴.

- $\mathcal{O}$  : seen, unseen 에 속하지 않는 클래스(background class)

---

**Algorithm 1** LAB algorithm

---

Given: annoData (annotated data), bgData (background/unannotated data),  $\mathcal{C}$  (set of all classes),  $\mathcal{S}$  (seen classes),  $\mathcal{U}$  (unseen classes),  $\mathcal{O}$  (background set), initModel (pre-trained network)

currModel  $\leftarrow$  train(initModel, annoData)

**for**  $i = 1$  to niters **do**

    currBgData  $\leftarrow \phi$

**for**  $b$  in bgData **do**

        // distribute background boxes over open vocabulary minus seen classes

$b_{new} \leftarrow \text{predict}(b, \text{currModel}, \mathcal{O})$

        //  $\mathcal{O} = \mathcal{C} \setminus (\mathcal{S} \cup \mathcal{U})$

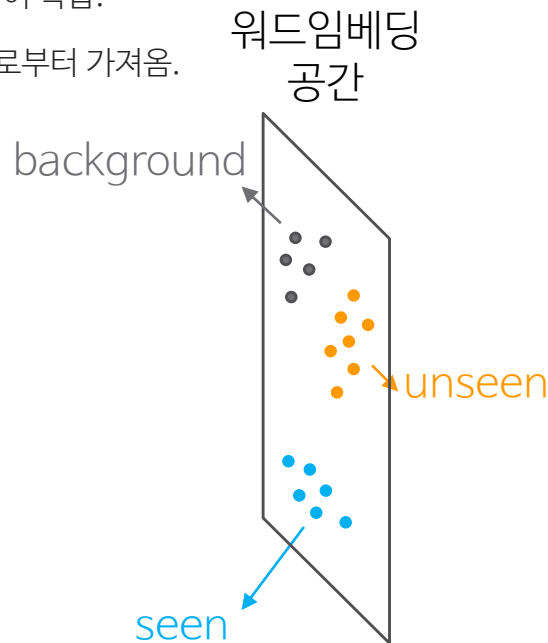
        currBgData  $\leftarrow$  currBgData  $\cup \{b_{new}\}$

    currAnnoData  $\leftarrow$  annoData  $\cup$  currBgData

    currModel  $\leftarrow$  train(currModel, currAnnoData)

**return** currModel

---



# Approach

## ▪ Background-aware zero-shot detection

### Algorithm 1 LAB algorithm

Given: annoData (annotated data), bgData (background/unannotated data),  $\mathcal{C}$  (set of all classes),  $\mathcal{S}$  (seen classes),  $\mathcal{U}$  (unseen classes),  $\mathcal{O}$  (background set), initModel (pre-trained network)

currModel  $\leftarrow$  train(initModel, annoData) // seen data로 baseline ZSD 학습

for  $i = 1$  to niters do

    currBgData  $\leftarrow \phi$

    for  $b$  in bgData do

        // distribute background boxes over open vocabulary minus seen classes

$b_{new} \leftarrow \text{predict}(b, \text{currModel}, \mathcal{O})$  // 임의의 background 바운딩 박스에 대한 predict 결과를 seen class가 아닌 클래스( $\in \mathcal{O}$ )에 할당

        //  $\mathcal{O} = \mathcal{C} \setminus (\mathcal{S} \cup \mathcal{U})$

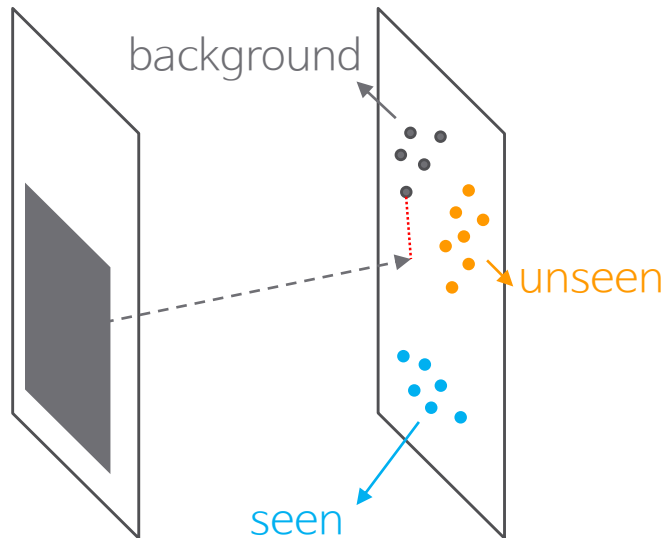
        currBgData  $\leftarrow$  currBgData  $\cup \{b_{new}\}$  // 현재 background class set에 추가

    currAnnoData  $\leftarrow$  annoData  $\cup$  currBgData

    currModel  $\leftarrow$  train(currModel, currAnnoData) // 다양한 background class가 포함된 데이터로 재학습

return currModel

- seen/unseen class 에 포함되지 않는 클래스를 학습에 참여시킴으로써
- 더 다양한 시각적 특징들을 좀더 학습할 수 있음.



- **Densely Sampled Embedding Space(DSES)**

- 본 논문에 소개된 ZSD는 label embedding space와 feature 로부터 projection 된 벡터가 common embedding space를 가지는 방법에 의존적
- 여기서 문제는 seen class의 개수가 많지 않을 경우, 임베딩 스페이스가 sparse 하여 학습이 제대로 안된다는 점.
- 따라서 seen class의 개수가 적으니 외부 데이터를 활용하여 임베딩 스페이스를 dense 하게 만들어 학습 성능을 높이는 방법을 적용.
- 실험 부분에서 디테일 설명.

- Summary

- Baseline Zero-Shot Detection(ZSD)

- max margin loss 를 최적화하여 (이미지→워드임베딩) 프로젝션 매트릭스 학습
    - 코사인 유사도로 predict

- Background-aware ZSD

- (1) Statically assigned background (SB)
      - 하나의 고정된 background vector 를 학습에 포함.
    - (2) Latent assignment based(LAB)
      - seen/unseen 에 포함되지 않는 여러 클래스를 background 클래스로 선택해 학습에 포함.

- Densely Sampled Embedding Space(DSES)

- seen class가 많지 않은 경우 임베딩 공간이 sparse해서 학습이 잘 안되므로 학습 성능을 높이기 위해 외부 데이터 활용

# Experiments

---

# Experiments

참조 - <http://cocodataset.org/#home>

## ▪ Datasets

### ▪ MSCOCO (Common Objects in COntext)

- MS에서 제공하는 이미지 데이터셋

- large-scale object dataset

- detection

- segmentation

- captioning

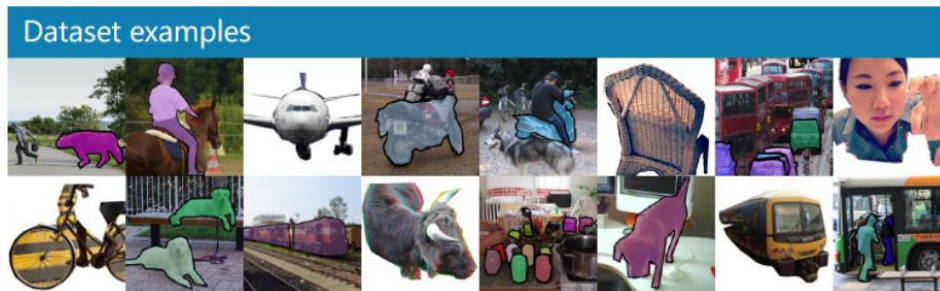
- object class : 80 개

- 이미지당 5개의 caption 제공

- 본 논문에서 학습에 사용한 label

- seen class : 48 개

- unseen class : 17개



a computer and speakers on a wood desk  
a computer screen and keyboard on a desk  
a white and black cocker spaniel under a computer table  
a dog that is laying down under a computer desk  
a dog laying under a brown computer desk.



a dog laying down with a cushion over its head.  
there is a dog playing with a dog bed  
a dog laying down with its head on its head  
a brown dog laying on floor under a brown mat  
a dog under a dog bed on a white background



# Experiments

참조 - <https://visualgenome.org/>

## ▪ Datasets

### ▪ VisualGenome (VG)

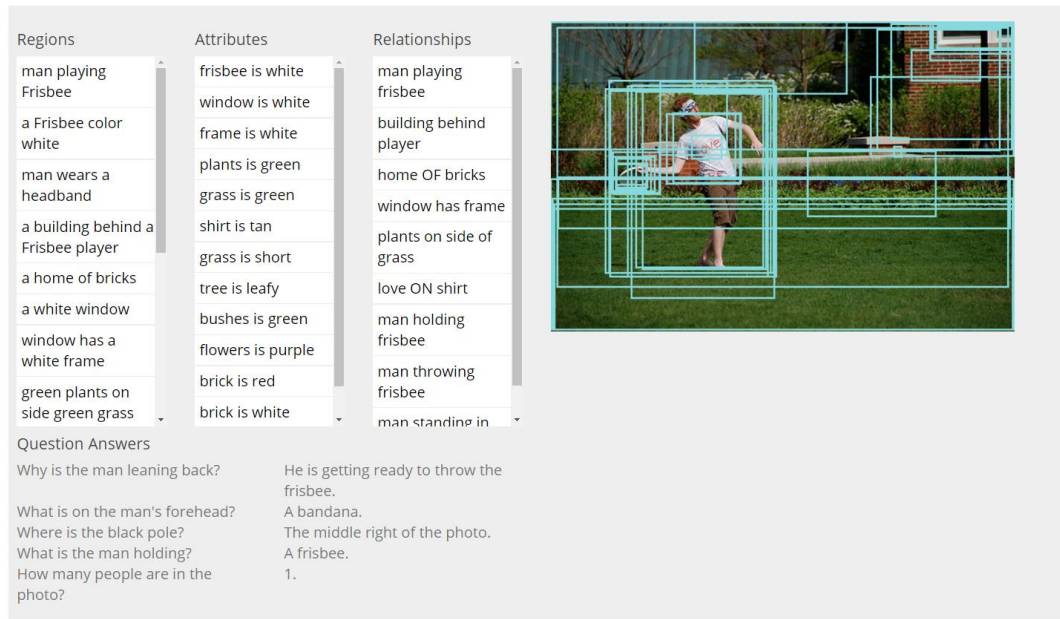
- 이미지와 더불어 이미지에 있는 오브젝트들의 바운딩 박스 정보 등의 다양한 annotation을 제공하는 데이터셋
- multi object vision task 에 주로 사용됨.
- object label 의 종류가 매우 다양함.
- WordNet synset으로 라벨링 되어있음.

### ▪ annotations

- object region(바운딩 박스)
- object label(각 바운딩 박스)
- object-object relationships
- attributes
- question-answers

### ▪ 본 논문에서 학습에 사용한 label

- seen class : 478 개
- unseen class : 130 개



The screenshot displays the Visual Genome interface for an image of a man playing frisbee. The interface is divided into several sections:

- Regions:** A list of object regions with their corresponding bounding boxes. Examples include "man playing Frisbee", "a Frisbee color white", "man wears a headband", "a building behind a Frisbee player", "a home of bricks", "a white window", "window has a white frame", "green plants on side green grass", and "man standing in".
- Attributes:** A list of attributes for the objects. Examples include "frisbee is white", "window is white", "frame is white", "plants is green", "grass is green", "shirt is tan", "grass is short", "tree is leafy", "bushes is green", "flowers is purple", "brick is red", and "brick is white".
- Relationships:** A list of relationships between objects. Examples include "man playing frisbee", "building behind player", "home OF bricks", "window has frame", "plants on side of grass", "love ON shirt", "man holding frisbee", "man throwing frisbee", and "man standing in".
- Question Answers:** A section for question-answer pairs. Examples include "Why is the man leaning back?" (He is getting ready to throw the frisbee.), "What is on the man's forehead?" (A bandana.), "Where is the black pole?" (The middle right of the photo.), "What is the man holding?" (A frisbee.), and "How many people are in the photo?" (1).

On the right side of the interface, there is a photograph of a man in a white shirt and dark shorts, standing on a grassy field, holding a frisbee. The image is overlaid with numerous cyan bounding boxes representing the detected regions.



# Experiments

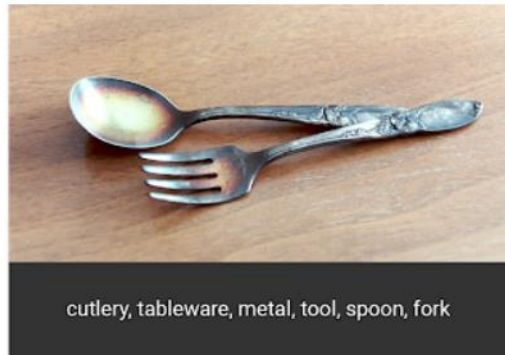
참조 - <https://opensource.google/projects/open-images-dataset>

## ▪ Datasets

### ▪ OpenImages (OI)

- 구글에서 제공하는 이미지 데이터셋
- 약 900만 장 이상의 이미지 URL
- object class : 6000개 이상
- 평균적으로 이미지당 8개 정도의 label 이 붙어 있음.

- 본 논문에서는 DSES 를 위해 활용



# Experiments

---

## ▪ Implementation detail

### ▪ dataset 준비

- background 바운딩 박스는 학습 데이터의 각 이미지에서 바운딩 박스 proposal 을 먼저 구함.
- 각 바운딩 박스 proposal 은 seen class 에 할당하거나 'background' class 에 할당됨.
  - ground truth와 IoU 측정
  - 거의 모든 바운딩 박스가 background 이므로 일부를 제외. 기준은  $0 < \text{IoU} < 0.2$  만 사용
  - 이와 별개로  $\text{IoU} = 0$  인 부분을 random sample하여 보충
  - $\text{IoU} > 0.5$  인 바운딩 박스는 ground truth의 클래스로 할당
- baseline ZSD 를 학습할 때는 seen class 만 사용하여 학습.

### ▪ Dense Sampling of Semantic Space

- MSCOCO와 VisualGenome 데이터셋을 보충하기 위해 OpenImages 데이터셋에서 test class(unseen class)를 제외한 클래스들과 이미지 바운딩 박스들을 train class 에 추가시킴.

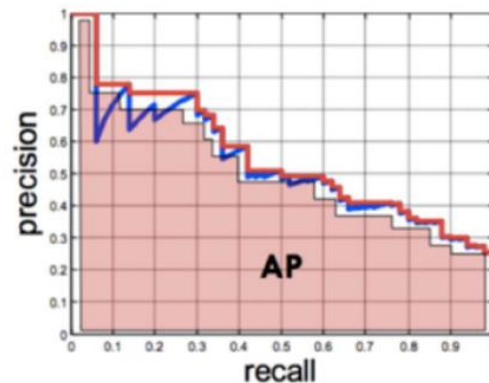
# Experiments

## ▪ Implementation detail

### ▪ Evaluation protocol

- 보통 detection task 에서 성능 평가 지표로 주로 사용하는 mAP 대신 recall 을 주요 성능 평가 지표로 삼음.
- 왜냐면 VG 데이터셋 같은 이미지 한장에 엄청나게 많은 오브젝트가 들어있는 데이터셋을 사용할 경우, 모든 객체를 라벨링하는 것은 거의 불가능함.
- 이런 데이터셋의 경우 detector 가 물체를 잘 찾았더라도 ground truth labeling 이 안되어있는 경우가 있음.
- mAP 로 평가를 하면 이런 것들을 전부 false positive 로 인식하여 성능을 제대로 보여주지 못함.
- 따라서 본 논문에서는 recall 을 사용함.

- Recall : 마땅히 검출해야 하는 물체 중 (TP + FN) 제대로 검출한 비율 (TP)
- Precision 모든 검출 결과 중 (TP + FP) 옳게 검출한 비율 (TP)



# Experiments

## Quantitative Result

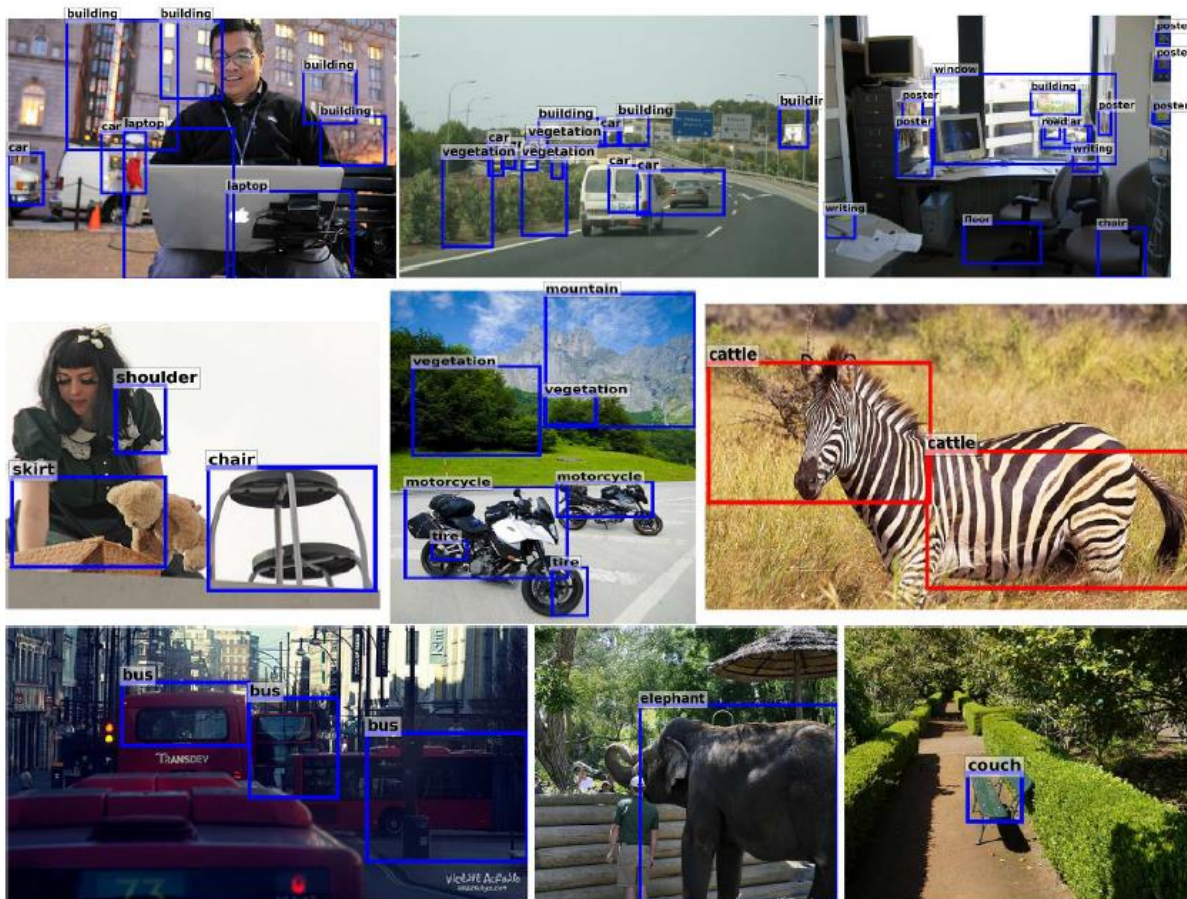
MSCOCO								Visual Genome					
ZSD Method	BG-aware	#classes			IoU			#classes			IoU		
		$ \mathcal{S} $	$ \mathcal{U} $	$ \mathcal{O} $	0.4	0.5	0.6	$ \mathcal{S} $	$ \mathcal{U} $	$ \mathcal{O} $	0.4	0.5	0.6
Baseline		48	17	0	34.36	22.14 (0.32)	11.31	478	130	0	8.19	5.19	2.63
SB	✓	48	17	1	34.46	24.39 (0.70)	12.55	478	130	1	6.06	4.09	2.43
DSES		378	17	0	<b>40.23</b>	<b>27.19</b> (0.54)	<b>13.63</b>	716	130	0	7.78	4.75	2.34
LAB	✓	48	17	343	31.86	20.52 (0.27)	9.98	478	130	1673	<b>8.43</b>	<b>5.40</b>	<b>2.74</b>

MSCOCO						
	Baseline			SB		
K↓ IoU→	0.3	0.4	0.5	0.3	0.4	0.5
All	47.91	37.86	24.47 (0.22)	43.79	35.58	<b>25.12</b> (0.64)
100	43.62	34.36	22.14 (0.32)	42.22	<b>34.46</b>	<b>24.39</b> (0.70)
80	41.69	32.64	21.01 (0.38)	41.47	<b>33.98</b>	<b>24.01</b> (0.72)
50	36.19	27.37	17.05 (0.50)	<b>39.82</b>	<b>32.6</b>	<b>23.16</b> (0.81)

VisualGenome					
Baseline			LAB		
0.3	0.4	0.5	0.3	0.4	0.5
13.88	9.98	6.45	12.75	9.61	6.22
11.34	8.19	5.19	11.20	<b>8.43</b>	<b>5.40</b>
10.41	7.55	4.75	<b>10.45</b>	<b>7.86</b>	<b>5.06</b>
7.98	5.79	3.68	<b>8.54</b>	<b>6.44</b>	<b>4.14</b>

# Experiments

## ▪ Qualitative Result



- Summary

- Dataset

- MSCOCO

- 80개 클래스, 이미지당 5개 captioning, segmentation 정보 제공.

- VisualGenome(VG)

- 다양한 multi-object가 포함된 이미지와 region, label, relationship, attribute, Q&A 등 annotation 제공

- OpenImages(OI)

- 임베딩 스페이스를 dense하게 해주기 위해 위 두 데이터셋에 포함된 seen class 를 보충하는 데에 이용함.

# Conclusion

---

- 이 논문의 contribution

- (1) Zero shot detection (ZSD) task 소개 및 baseline ZSD 제안

- (2) 이미지의 background 부분의 정보를 활용하여 background-aware detector를 학습하기 위한 2가지 방법을 소개

- (1) Static Background (SB)

- (2) Latent assigned background (LAB)

- (3) 부가적인 데이터를 이용하여 임베딩 스페이스를 dense 하게 해주면 클래스 수가 적은 데이터셋을 제로샷 러닝으로 학습할 때 도움이 됨.

감사합니다

---



# Reference

---

- Paper
  - [http://openaccess.thecvf.com/content\\_ECCV\\_2018/papers/Ankan\\_Bansal\\_Zero-Shot\\_Object\\_Detection\\_ECCV\\_2018\\_paper.pdf](http://openaccess.thecvf.com/content_ECCV_2018/papers/Ankan_Bansal_Zero-Shot_Object_Detection_ECCV_2018_paper.pdf)
- 논문 저자의 논문 소개
  - 홈페이지 : <http://ankan.umiacs.io/zsd.html>
  - blog post : <https://computervision.blogspot.com/2018/04/zero-shot-object-detection.html>
- Zero Shot Learning 개념 설명 관련
  - 영상 강의(1) <https://youtu.be/0iKsimVvfjE>
  - 영상 강의(2) <https://youtu.be/dE4nU5OaQqA>
  - 영상 강의(3) <https://youtu.be/jBnCCR-3bXc>
- 관련 연구
  - 2018, ACCV : [https://salman-h-khan.github.io/ProjectPages/ZSD\\_Arxiv18.html](https://salman-h-khan.github.io/ProjectPages/ZSD_Arxiv18.html)