

# DEEP LEARNING BASED COMPUTER VISION

---

최우빈

# 진행 과정

---

- 사용 Tool : Python
- 사용 알고리즘 : CNN(합성곱 신경망)
- 주제 : SSD(Single Shot multibox Detector)
- 차량용 블랙박스 영상 기반의 Object Detection(객체 인식)

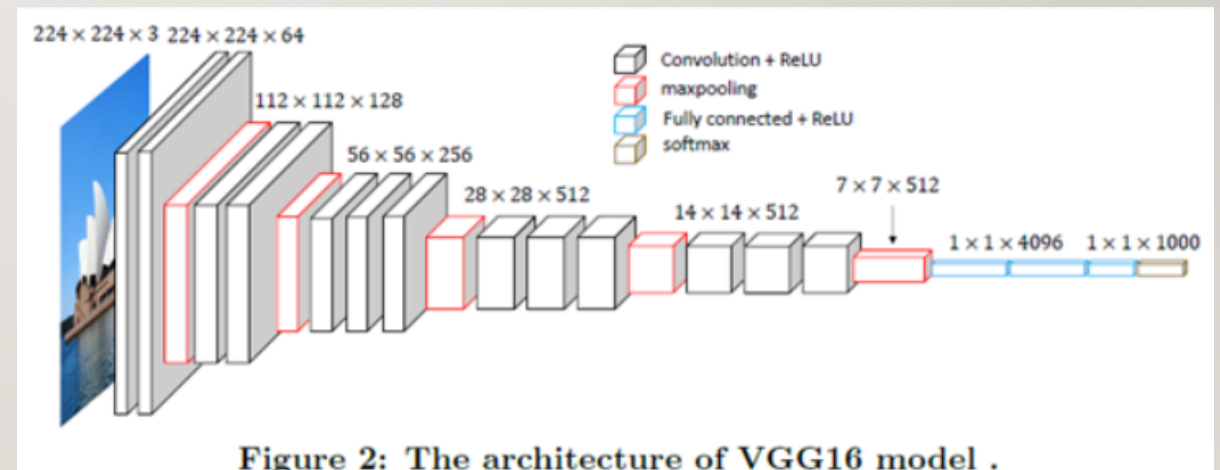
# WHAT IS SSD?

---

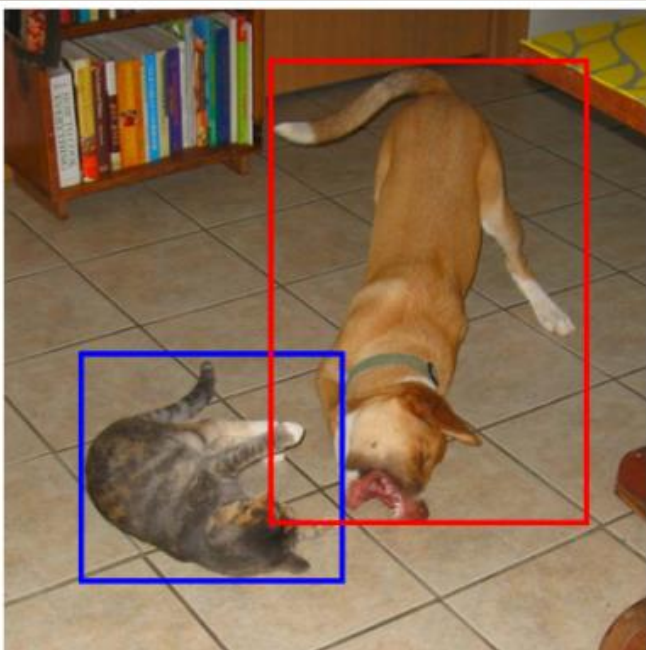
## Single-shot detector는 사진의 변형 없이 그 한 장으로 훈련, 검출을 하는 detector를 의미

## 보편적으로 사용되는 Deep Learning Detector : Alex net, VGG-16 net

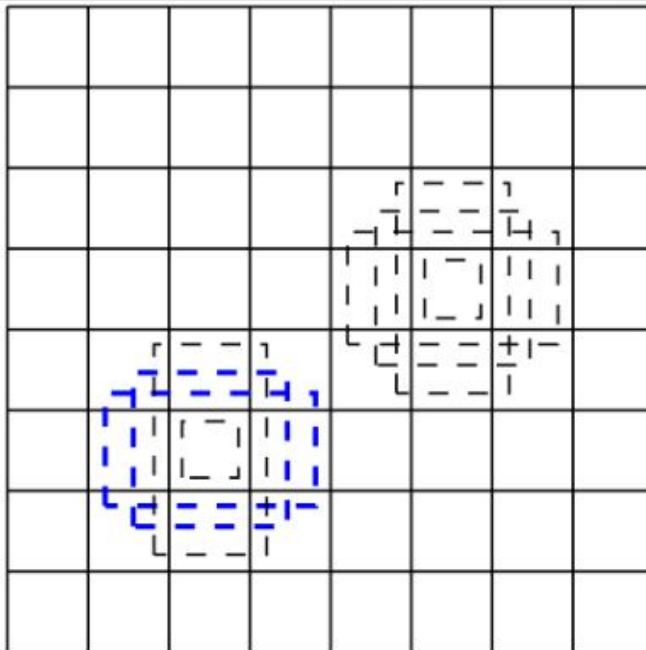
## 124 X 124 크기의 이미지를 입력으로 받아, 결과를 1000 Labels에 대한 확률로 반환



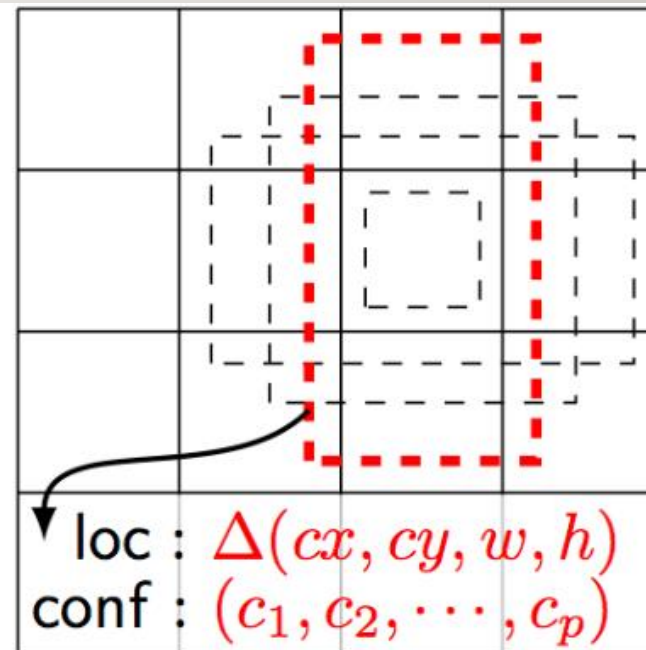
# MULTI SCALE FEATURE MAPS FOR DETECTIONS



(a) Image with GT boxes



(b)  $8 \times 8$  feature map

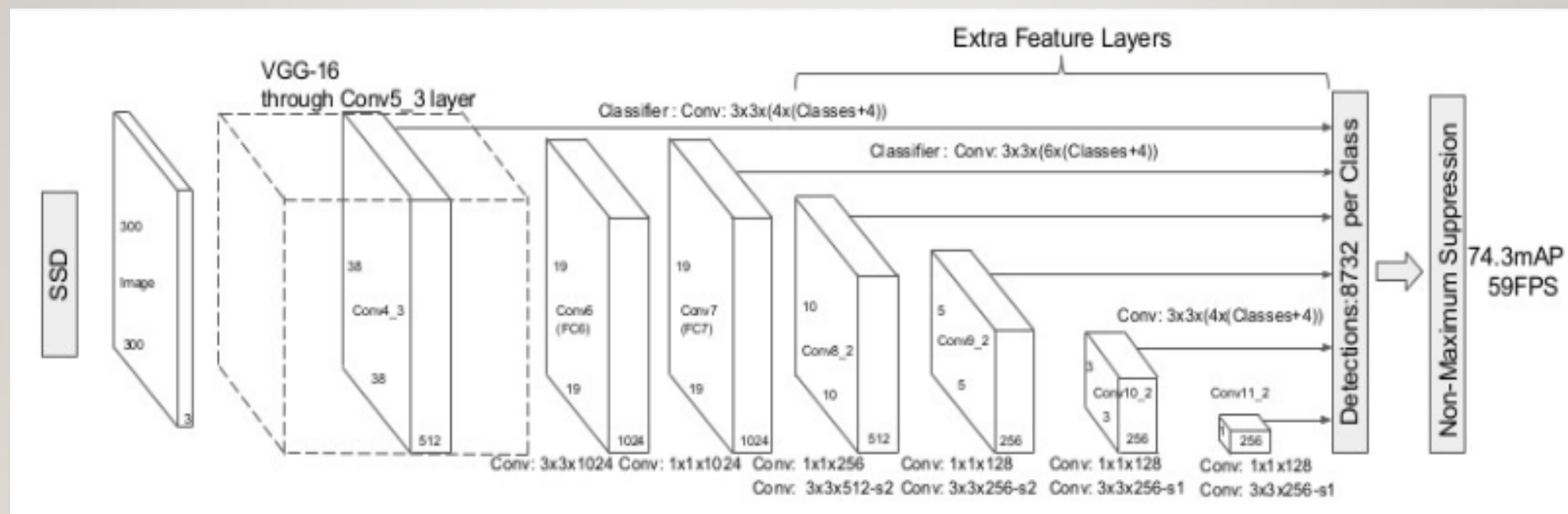


loc :  $\Delta(cx, cy, w, h)$   
conf :  $(c_1, c_2, \dots, c_p)$

(c)  $4 \times 4$  feature map



# SSD ARCHITECTURES



- 
- bounding box offset : 각 cell(feature map 한 칸)을 기준으로 한 상대적 위치와 박스의 크기
  - Matching Strategy : SSD가 예측한 박스와 실제 박스가 일치하는 지를 확인하는 것은 매우 중요한 문제이다. 이럴 때 가장 많이 사용하는 것은 jaccard overlap, 혹은 intersection over Union(IOUS) 이다. SSD에서는 이 값이 일정 값(threshold, 본 논문에서는 0.5)를 넘기기만 하면 일단 일치한다고 가정

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

- 
- threshold를 넘는 상자를 모두 선택하는 이유는 높은 정확도를 가진 상자를 한꺼번에 여러 번 학습시킴으로써 하나만 고르는 것보다 높은 학습율을 얻기 위해서

# TRAINING OBJECTIVE

---

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

$$L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx}) / d_i^w \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy}) / d_i^h$$

$$\hat{g}_j^w = \log \left( \frac{g_j^w}{d_i^w} \right) \quad \hat{g}_j^h = \log \left( \frac{g_j^h}{d_i^h} \right)$$

$$L_{conf}(x, c) = - \sum_{i \in Pos} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$





## SCALE & ASPECT RATIOS FOR DEFAULT BOXES

---

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m]$$

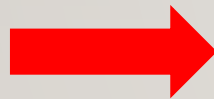
$s_{\min} = 0.2$  ,  $s_{\max} = 0.9$

$m$  : number of multi feature maps

- 
- 훈련 데이터 : 차량 관련 데이터 와 차량과 관련 없는 데이터(신호등, 교통 표지판, 도로 등)로 부터 훈련  총 34개 카테고리 분류
  - CNN으로 부터 각 layer마다 파라미터 입력 후 모델링  
 Tensorflow – gpu 라이브러리 이용하여 훈련 (GTX1050 GPU 이상 권장)
  - train set : 22136 images
  - Validation set(자체 검증) : 4952 images

- 
- 기본 video.mp4를 입력으로 넣고 자체 검증 데이터 셋으로부터 정확도 측정  
(객체 인식을 한 video.mp4 형식으로 저장)

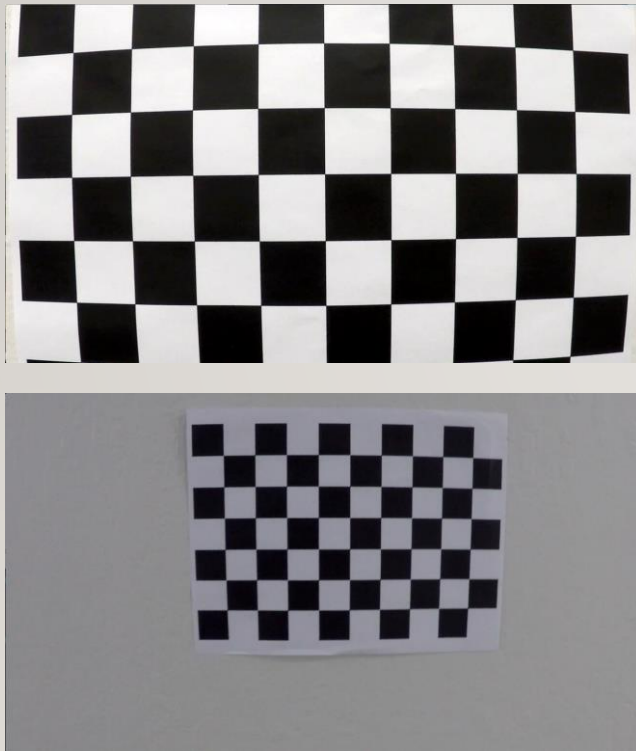
## 기본 형식이 없더라도, 원하는 파일을 YOUTUBE에서 검색 후 pytube 라이브러리를 이용하여 저장 가능

 변환과정 : 기본 video.mp4를 video file clip을 이용하여 이미지로 변환하여 객체 인식 후, 변환 video.mp4 형식으로 저장

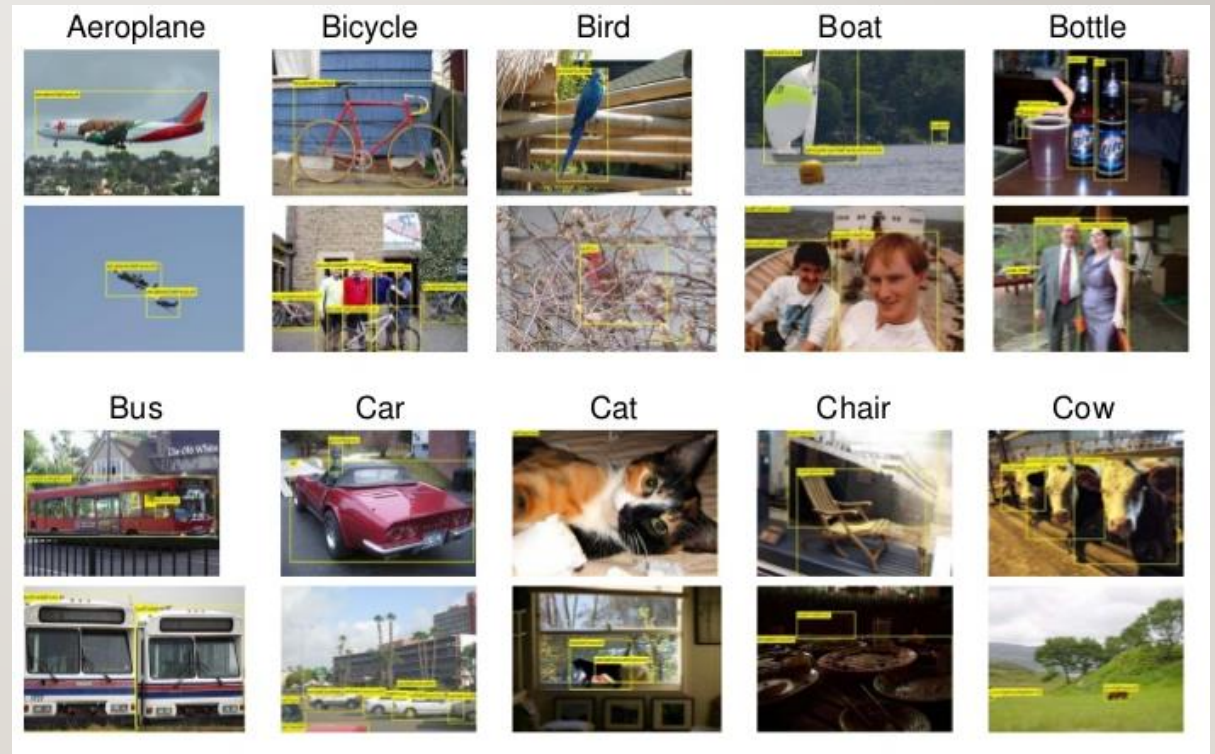
# LANE DETECTION & OBJECT DETECTION DATA SET

---

Road data



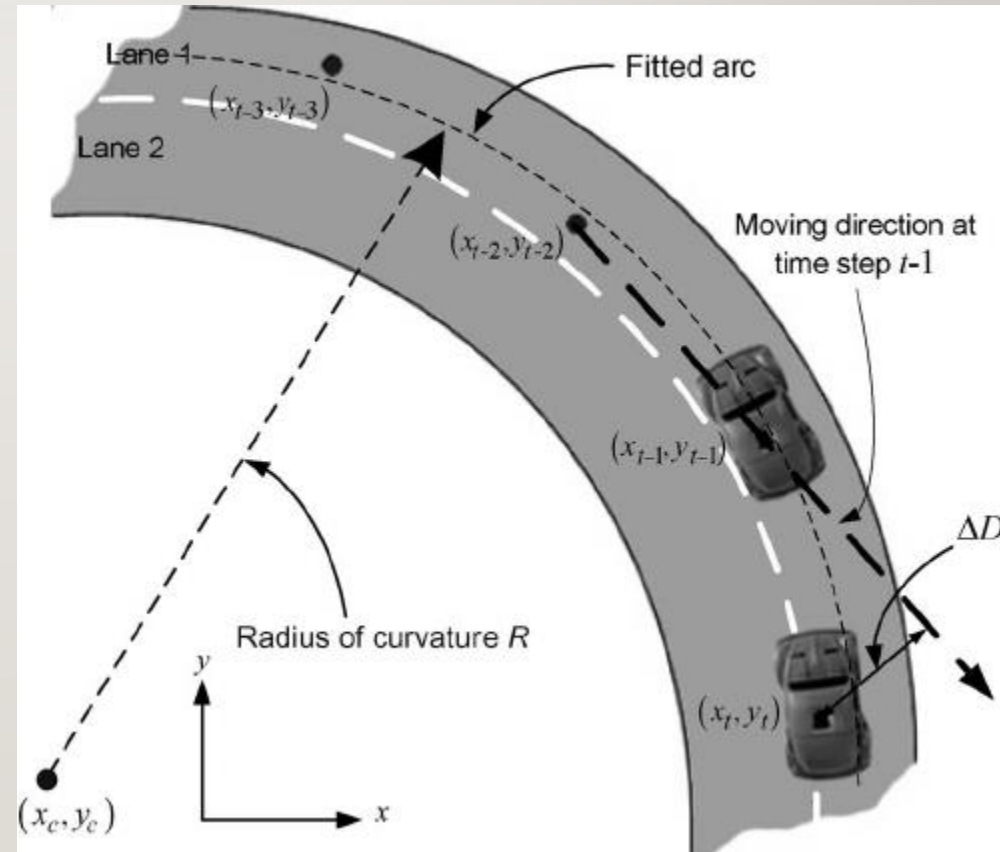
Object data





# 곡률 반경

Radius of Curvature을 함수로 입력하여  
객체 인식한 mp4 파일에 실시간으로  
거리를 도출



# SOURCE CODE & RELATED DOCUMENTS

---

[https://github.com/choiwb/ADAS\\_Detection](https://github.com/choiwb/ADAS_Detection)