📖

[논문리뷰] YOLOv1 [2016]

✨ 제가 **제일 좋아하는 논문 중 하나**입니다. 상세하게 작성해보도록 하겠습니다 💞

Introduction

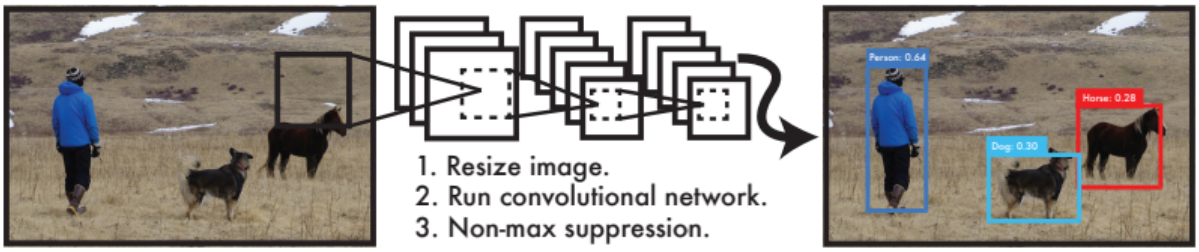
✨ **YOLO(You Only Look Once)** 한번에 보고 한번에 처리한다구!!!

> We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are.

YOLO는 Object Detection에서 새로운 접근방식을 제시합니다.
Object Detection Problem에서 spatially seperated bounding boxes와 associated class probabilites의 **Regression Problem으로 frame하였습니다.**

YOLO의 unifed architecture는 매우 빠릅니다. **다른 SOTA Detection System에 비해, Localization error를 가지고 있지만, Background에 대한 False Positives는 상당히 낮습니다.**

YOLO는 매우 일반화된 객체의 대표성을 배운다.
신경망 이미지에서부터 예술작품과 같은 다른 도메인으로 일반화 할 때, DPM과 R-CNN 객체인지 방법을 능가합니다.



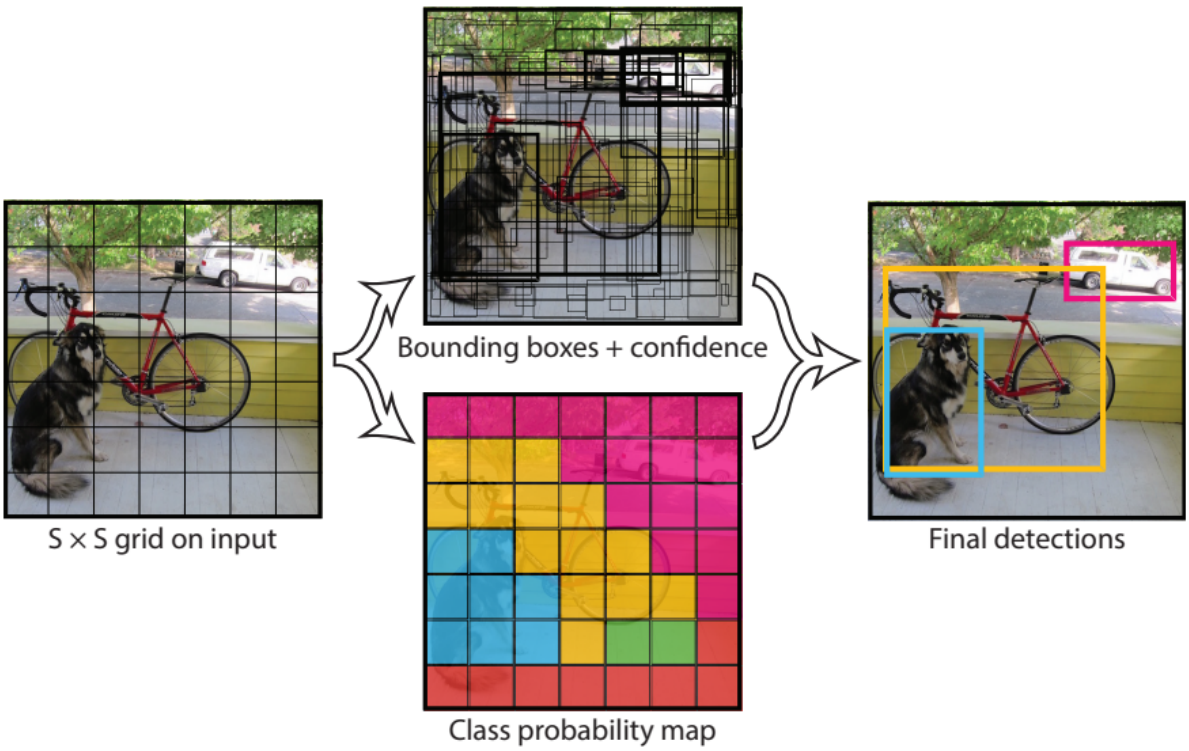
하나의 Convolutional Network가 여러 Bounding Box와 Bounding Box의 class probabilities를 동시에 predict합니다.

이미지 전체를 학습하여, 곧바로 Detection Performance를 최적화합니다. 이러한 통합된 모델은 객체 탐지에서 상당한 Benefit이 있습니다.

Unified Detection

YOLO는 객체 검출의 개별 요소를 Single Neural Network으로 통합된 모델입니다.
Network는 entire image로부터 predict한 each bounding box를 활용하여, feature를 사용합니다.
이러한 **Architecture덕분**에 **high Accuracy를 유지**하면서, **end-to-end학습 과 real-time speeds로 객체 검출이 가능**합니다.

YOLO는 Input image를 S x S Grid로 나눕니다. 만약 **어떤 center of Object가 Grid Cell에 위치한다면, Grid Cell이 객체를 검출**해야 합니다.
각각의 그리드 셀(gird cell)은 B개의 Bounding Box와 Bounding Box에 대한 Confidence Score를 예측합니다. **Confidence Score는 Bounding Box가 객체를 포함**한다는 것을 얼마나 믿을만한지, 그리고 **예측한 Bounding Box가 얼마나 정확**한지 나타냅니다.



각각의 **Bounding Box는 5개의 예측치로 구성**되어 있습니다. **x, y, w, h, confidence**입니다.

(X, Y)좌표 쌍은 **Bounding Box 중심의 그리드 셀(Grid Cell) 내 상대 위치**를 뜻합니다.

각각의 **Grid Cell은 Conditional Class Probabilities(C)를 예측**합니다. **Conditional Class Probabilities**는 다음과 같이 **계산**할 수 있습니다.

이는 Grid Cell안에 객체가 있다는 조건 하에 그 객체가 어떤 Class인지에 대한 조건부 확률입니다.

Grid Cell에 몇 개의 Bounding Box가 있는지와는 무관하게 하나의 Grid Cell에는 오직 하나의 Class에 대한 확률 값만을 구합니다.

하나의 Grid Cell은 B개의 Bounding Box를 예측한다고 했습니다. B의 개수와는 무관하게 하나의 Grid Cell에서는 Class 하나만을 예측하는 것입니다.

테스트 단계에서는 conditional class probability(C)와 개별 boudning box의 confidence score를 곱해주는데, 이를 각 bounding box에 대한 class-specific confidence score라고 부릅니다. class-specific confidence score는 다음과 같이 계산할 수 있습니다. 위에서 구한 conditional class probability와 confidence score를 곱한 값입니다.

$$
\begin{aligned}
&class\ specific\ confidence\ score \\
&= \Pr(Class_i|Object) * \Pr(Object) * IOU_{pred}^{truth} \\
&= \Pr(Class_i) * IOU_{pred}^{truth}
\end{aligned}
$$

이 score는 **bounding box에 특정 클래스(class) 객체가 나타날 확률**(=Pr(Class_i))과 **예측된 bounding box가 그 클래스(class) 객체에 얼마나 잘 들어맞는지(fits the object)**(=IOU_pred^truth)를 나타냅니다. (These scores encode both the probability of that class appearing in the box and how well the predicted box fits the object.)

Unified Detection Summary

- YOLO는 객체 검출의 개별 요소를 **Single Neural Network로 통합한 모델**입니다.
- Input Images를 **S x S 그리드 (S x S grid)**로 나눕니다.
- 그리드 셀(grid cell)은 **B개의 Bounding Box**와 Bounding Box에 대한 **Confidence score** 예측
- **Class-specific Confidence Score**를 통하여 **Class가 얼마나 잘 들어맞는지 예측**한다.
- **최종 예측 텐서의 Dimension**은 **(7 x 7 x 30)**입니다.

Network Design & Architecture

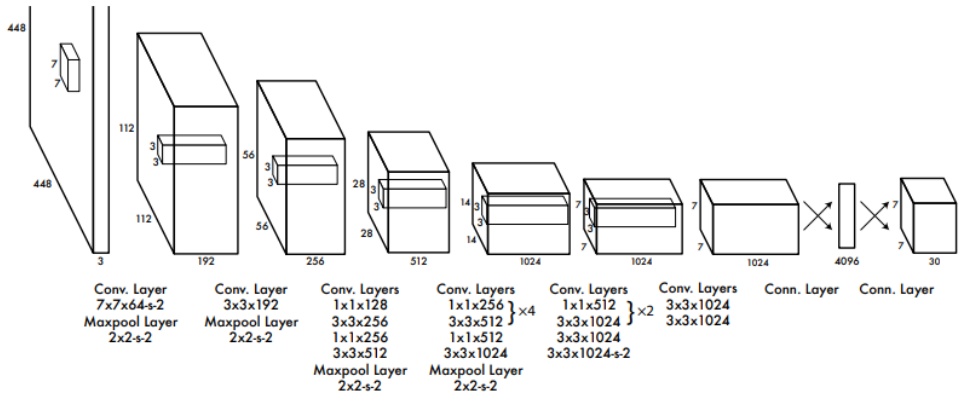


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Convolutional Layer은 Image로부터 feature를 extract하고, **fully connected layer은 클래스 확률**과 **Bounding Box의 Coordinates를 예측**합니다.

✨ **Final Output of Our Network is the 7 x 7 x 30 tensor of predictions.**

YOLO연구진은 훈련**(training)**과 추론**(inference)**을 위해 **DarkNet Framework를 사용**하였습니다.

Training Summary

- ImageNet Dataset으로 YOLO의 20개의 CNN을 pretrained합니다.
- pretrained 뒤에 4개의 Conv2d layer와 2개의 FC를 추가합니다.
- 마지막 layer는 linear activation function을 적용하고,
  나머지 모든 계층에는 leaky ReLU를 적용합니다.

Problem Improvement

- localization loss와 classification loss중 localization loss의 weights를 증가시킵니다.
- 객체가 없는 그리드 셀의 Confidence loss보다 객체가 존재하는 Grid Cell의 Confidence loss를 증가시킵니다.
- Bounding Box의 너비 와 높이에 **square root를 취해준 값을 loss function으로 사용**합니다.
  Overfitting을 막기 위해 Dropout과 Data Augmenatation을 사용합니다.

Inference

Just like in training, Test Image로부터 객체를 검출하는데, 하나의 신경망을 활용하면 됩니다.

한 **이미지당 98개의 Bounding Box를 예측**해주고, **Bounding Box마다 클래스 확률**을 구하면 됩니다.

Limitations of YOLO

YOLO는 하나의 Grid Cell마다, **두 개의 Bounding Box를 예측**합니다. 각각의 Grid Cell은 하나의 Object만을 검출할 수 있습니다. YOLO는 **strong spatial Constraints를 impose**합니다.

💡 **Spatial Constraints**

"하나의 Grid Cell은 오직 하나의 객체만 검출하므로, Grid Cell에 2개 이상의 객체가 있다면 객체 검출이 제한적입니다."

Finally, **YOLO모델**은 큰 Bounding Box와 작은 Bounding Box의 loss에 대해 동일한 Weights를 둔다는 단점이 있습니다. 크기가 큰 Bounding Box는 위치가 달라져도 비교적 성능에 별 영향을 주지 않지만, 크기가 작은 Bounding Box는 위치가 조금만 달라져도 성능에 큰 영향을 줄 수 있습니다.

큰 **Bounding Box에 비해, Bounding Box가 위치 변화에 따른 IOU변화가 더 심하기 때문입니다.**

Experiments

✨ Object Detection은 표준화된 **객체 검출 파이프라인을 만드는데 빠르게 초점을 두고 있습니다.**

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.

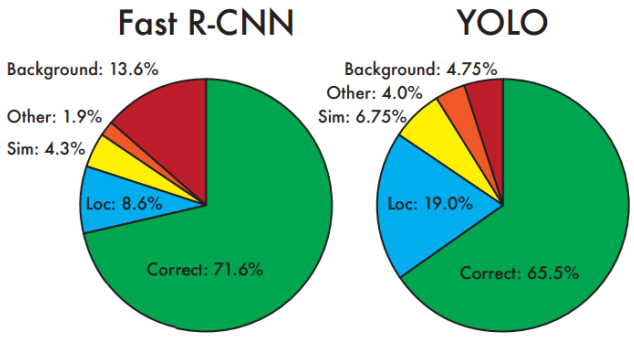


Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

YOLO는 localization error가 상대적으로 큽니다. localization error는 19.0%로 나머지 Error를 모두 합한 15.5%보다 큽니다. Fast R-CNN은 YOLO에 비해 Localization Error가 작습니다. **반면 Background Error가 상대적으로 큽니다. Fast R-CNN은 YOLO에 비해 Background Error가 3배나 더 큽니다.**

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	**73.9**	**85.5**	**82.9**	**76.6**	**57.8**	**62.7**	**79.4**	77.2	86.6	**55.0**	**79.1**	**62.2**	87.0	**83.4**	**84.7**	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	83.3	**81.8**	**48.6**	**73.5**	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	**89.4**	49.4	75.5	57.0	**87.5**	80.9	81.0	74.7	41.8	71.5	68.5	**82.1**	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [28]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	**68.8**	75.9	71.4
NoC [29]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [33]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

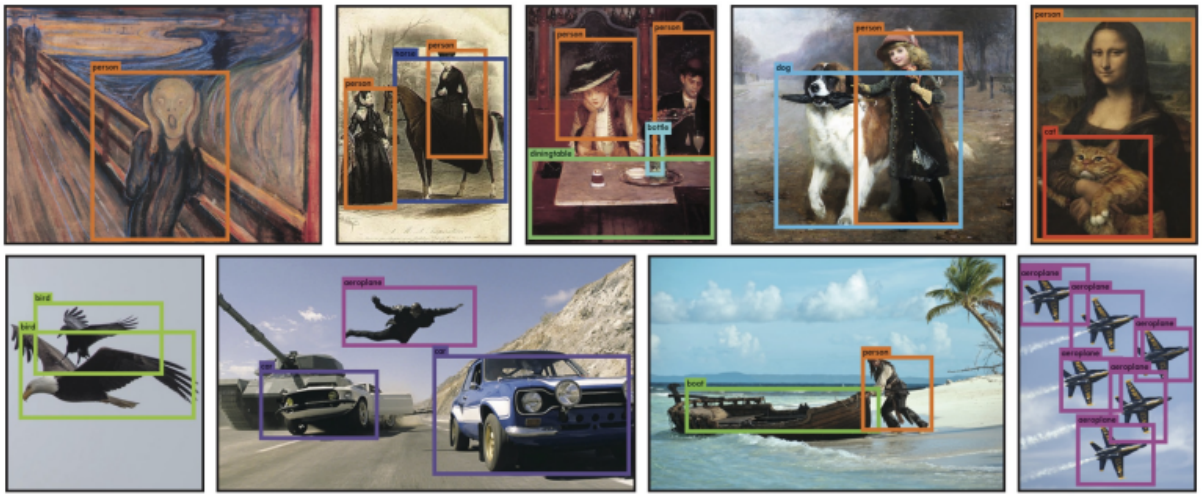
Table 3: PASCAL VOC 2012 Leaderboard. YOLO compared with the full comp4 (outside data allowed) public leaderboard as of November 6th, 2015. Mean average precision and per-class average precision are shown for a variety of detection methods. YOLO is the only real-time detector. Fast R-CNN + YOLO is the forth highest scoring method, with a 2.3% boost over Fast R-CNN.

속도 측면에서, YOLO가 빠르고, 정확도 측면에서는 **Fast R-CNN과 YOLO를 결합한 모델**이 좋습니다.

Real-Time Detection In The Whild

YOLO는 ComputerVision에서 활용할 수 있는 **빠르고 정확한 객체 검출 모델**입니다.

연구진은 **YOLO를 웹캠과 연결하여 실시간(real-time)객체 검출을 수행**하였습니다.



Conclusion

YOLO는 단순하면서도, 빠르고 정확합니다. 또한, YOLO는 훈련 단계에서 보지 못한 새로운 이미지에 대해서도 객체 검출을 잘합니다. 즉, **새로운 이미지에 대해서도 강건합니다.**

You Only Look Once: Unified, Real-Time Object Detection

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression

https://arxiv.org/abs/1506.02640

논문 리뷰 - YOLO(You Only Look Once) 톺아보기

본 글은 YOLO 논문 전체를 번역 및 설명해놓은 글입니다. 크게 중요하지 않은 부분을 제외하고는 대부분의 글을 번역했고 필요하다면 부가적인 설명도 추가했습니다. 내용이 긴 섹션 끝에는 요약도 추가했습

https://bkshin.tistory.com/entry/%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-YOLOYou-Only-Look-Once

YOLOv1 from Scratch

Oh boy. Hopefully this will leave you with a deep understanding of YOLO and how to implement it from scratch!Download Dataset here:https://www.kaggle.com/dat...

https://www.youtube.com/watch?v=n9_XyCGr-MI&ab_channel=AladdinPersson