Review

- **Transfer learning**: Change the last layer, initialize, and train for your application (relatively small amount of data)

- Object detection = (classification + localization) of multiple objects
- Bounding box proposal
  - Sliding window: Every location, every size
  - Selective search: Considering pixel relationship
- R-CNN (Great! But **super slow!**)
  - Selective search (~2000 boxes per image), Warping (227x227)
  - Feature extraction by using a **CNN** (transfer learning), **SVM** classifiers, **BB reg**
  - Non-max suppression
- SPPnet: BB on a CNN feature map, SPP layer for scale invariance
- Fast R-CNN: Single-stage training (softmax and multi-task loss)
- Faster R-CNN: RPN on GPU instead of selective search

One-stage 2D Object Detectors

Lecture 5

Hyung-Sin Kim

SNU Graduate School of Data Science

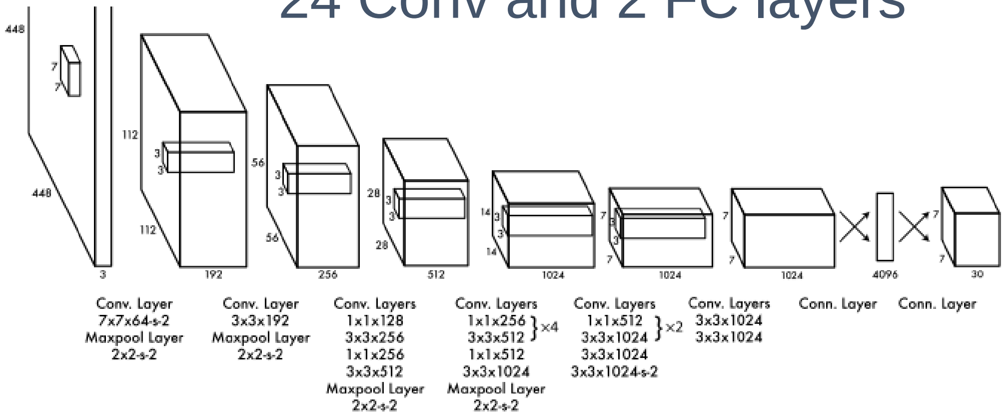
*Can we use a **single** module that gives bounding boxes for all the objects and also classify them by processing an entire image **only once**?*

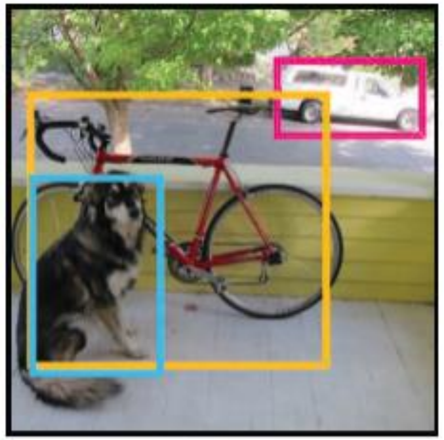
YoLo [CVPR'16] – You Only Look Once

- Insight: Humans glance at an image and instantly know what objects are in the image
- Process an entire image **only once**, instead of processing each bounding box
- Training **only one CNN** is enough to detect multiple objects in an image!

S × S grid on input

24 Conv and 2 FC layers



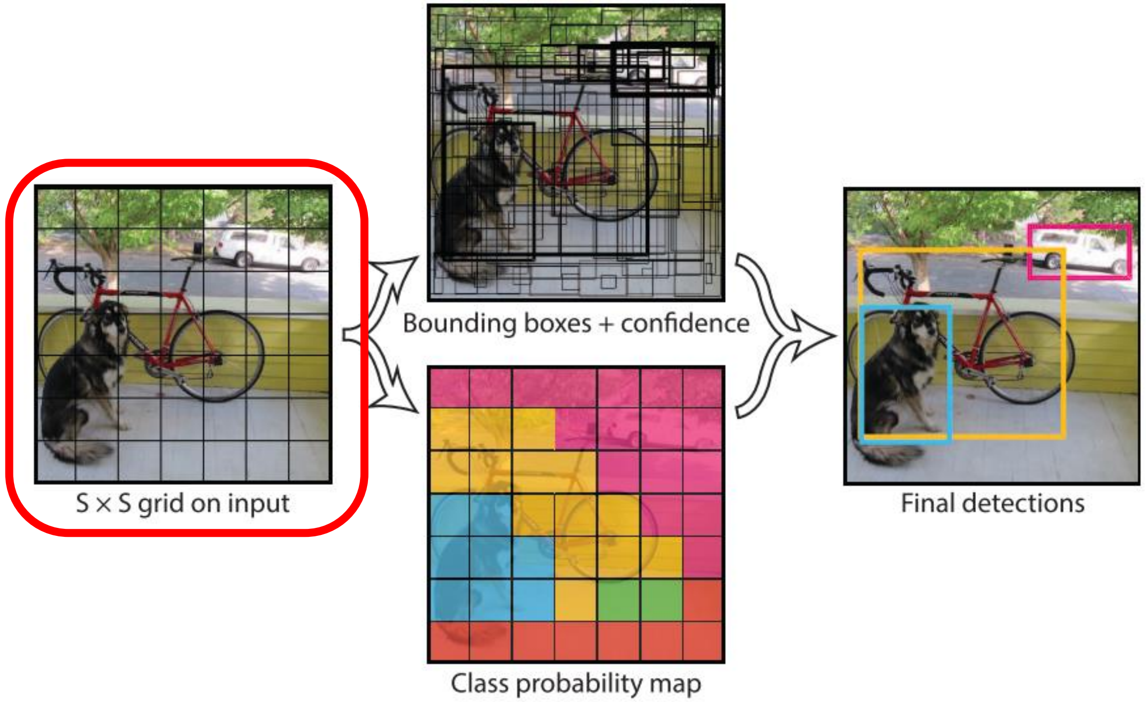


Final detections

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Detection Flow (Input)

- Divide an image into ***S x S*** grid



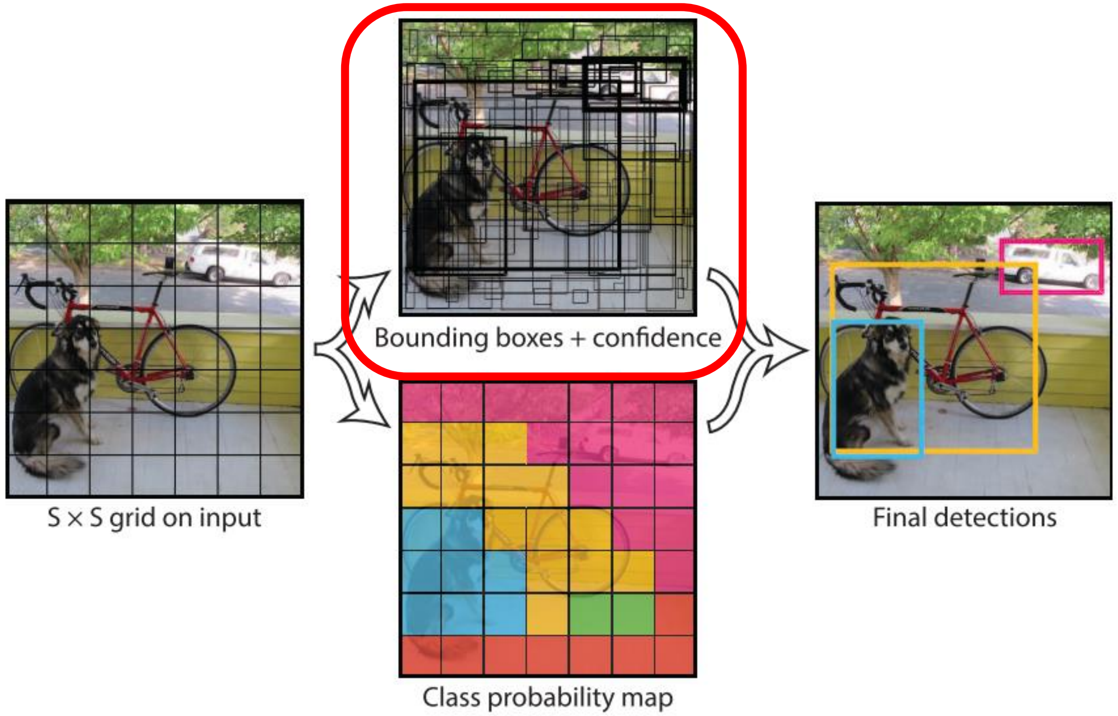
[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Detection Flow (Box & Score)

- Each grid cell proposes ***B*** bounding boxes
  - Each box has its center point in the grid cell
  - Bounding box representation

$$[p_c \quad b_x \quad b_y \quad b_h \quad b_w]$$

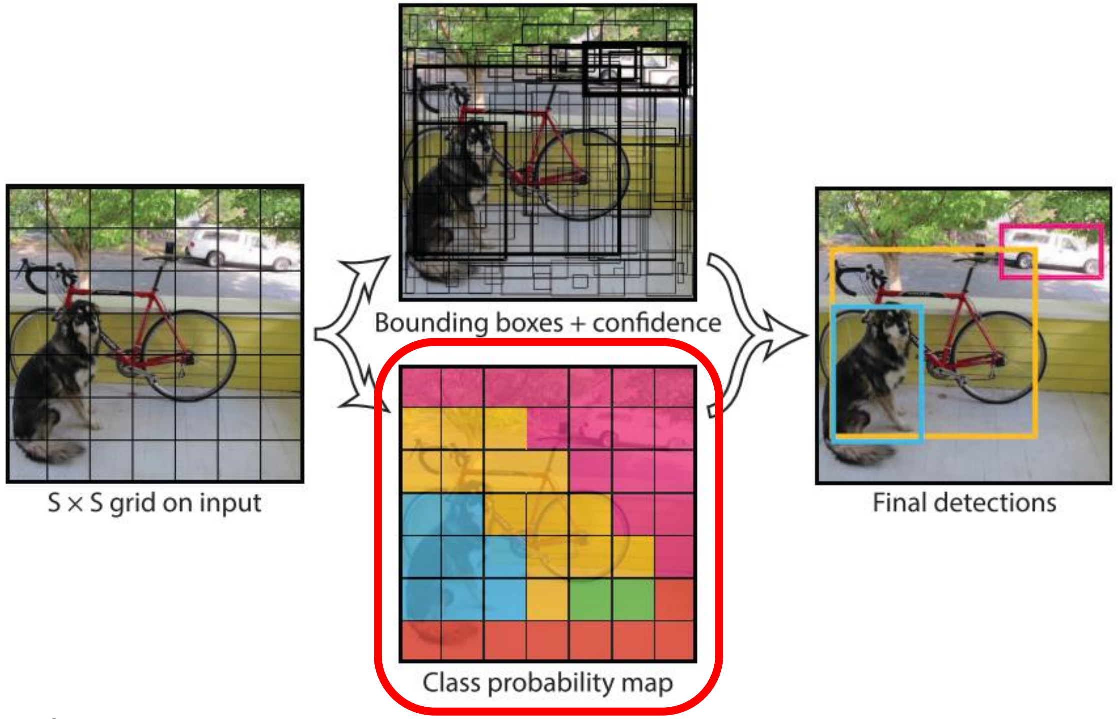
Confidence level of the bounding box's correctness
(1) containing a valid object
(2) accurate box location



[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Detection Flow (Box & Score)

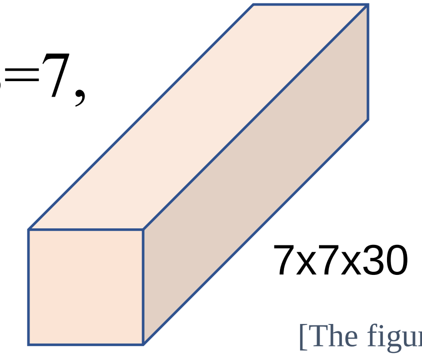
- Each cell proposes **one** class probability set
  - There are **C** classes
  - Predict conditional probability $P(C_i|\text{object})$ for each class ($1 <= i <= C$)
  - Result
    - $\{P(C_1|\text{object}), P(C_2|\text{object}), \cdots, P(C_C|\text{object})\}$
    - Sum of all the elements = 1
  - One cell proposes **multiple boxes** but **only one class**!

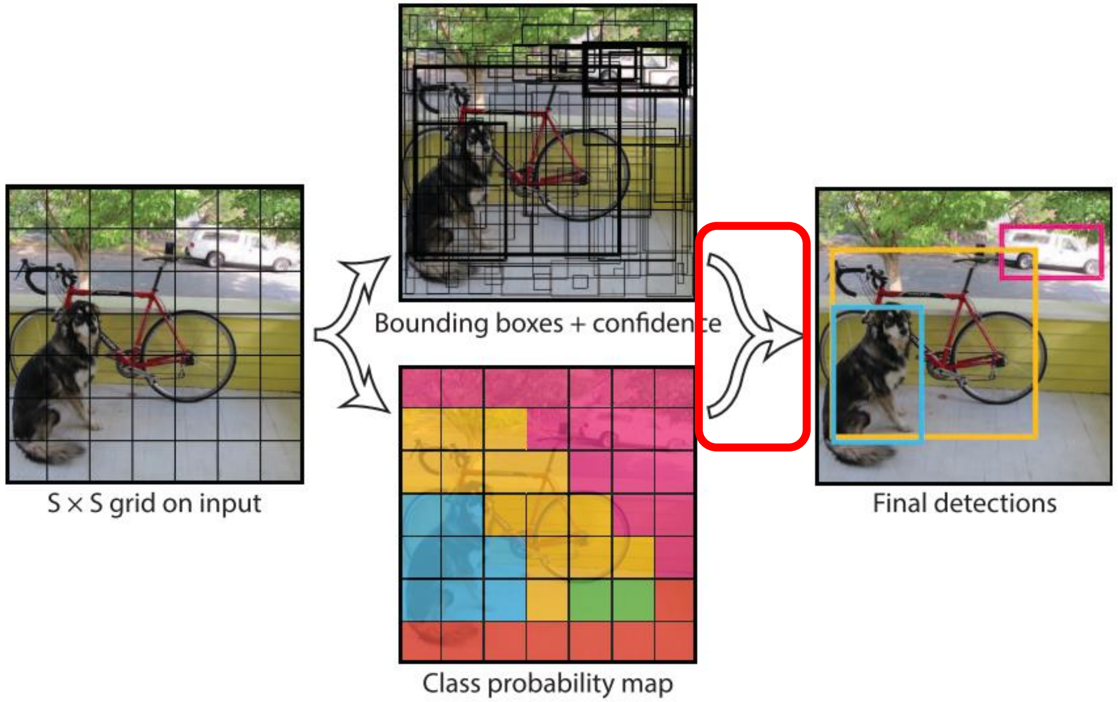


[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Detection Flow (Pred Metric)

- Now, each cell has a [5x**B** + **C**] vector
  - When B=2 and C=20, the vector has 30 entries
  - [$\boxed{p_{c1}, b_{x1}, b_{y1}, b_{h1}, b_{w1},}$ Bounding box 1
    $\boxed{p_{c2}, b_{x2}, b_{y2}, b_{h2}, b_{w2},}$ Bounding box 2
    $\boxed{\text{ClassProbabilitySet}}$] 20 prob values
- The whole image has an S x S x (5xB+C) tensor
  - When B=2, C=20, and S=7, this is a 7x7x30 tensor



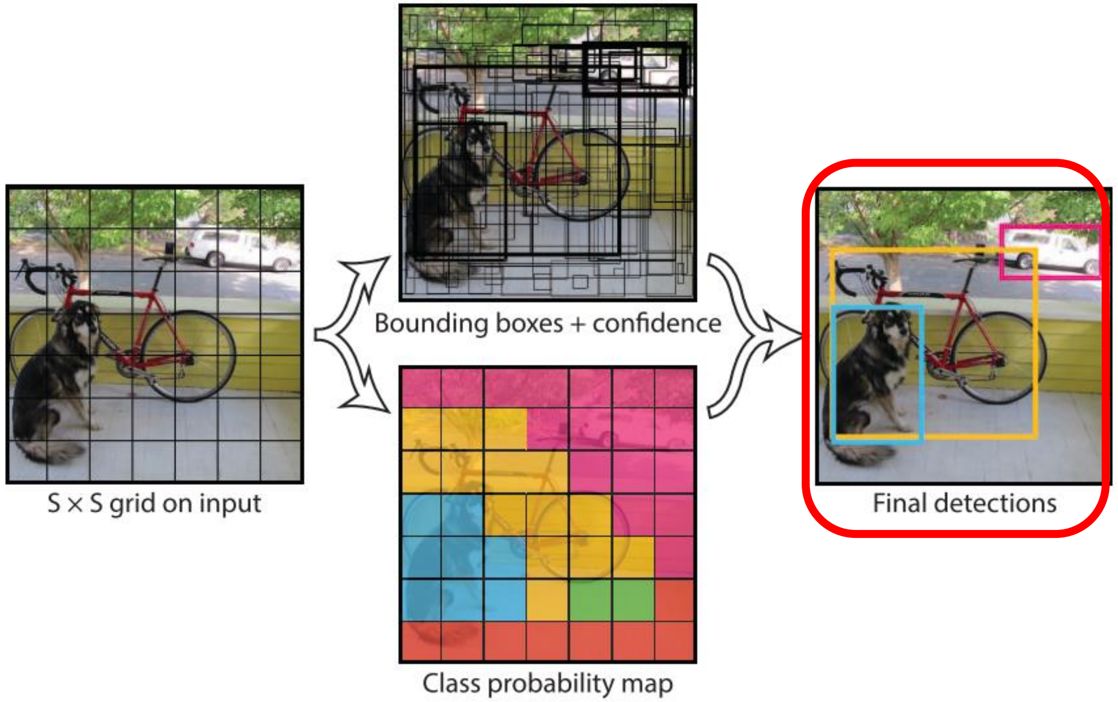


[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Detection Flow (Refining)

- Get rid of bounding boxes with low confidence
  - Now each remaining box is sure that it contains an object and knows what it is

- Non-max suppression for each class
  - One box for one object!

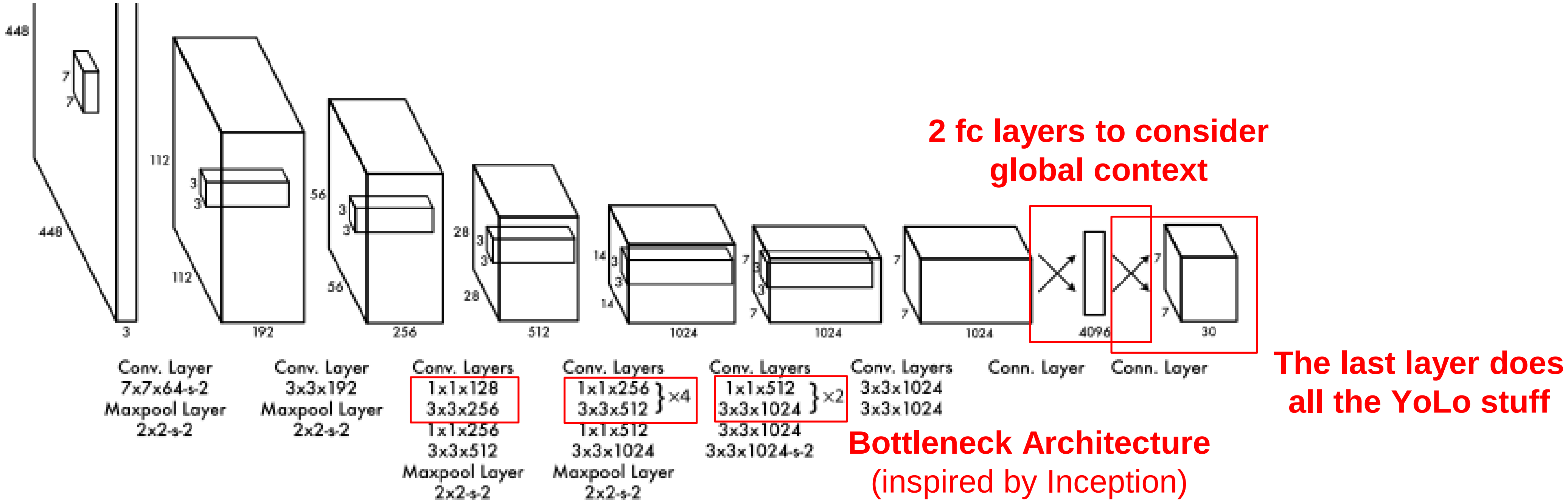


[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

Let's train ***a single CNN*** *to do all these things!*

YoLo [CVPR'16] – CNN Architecture

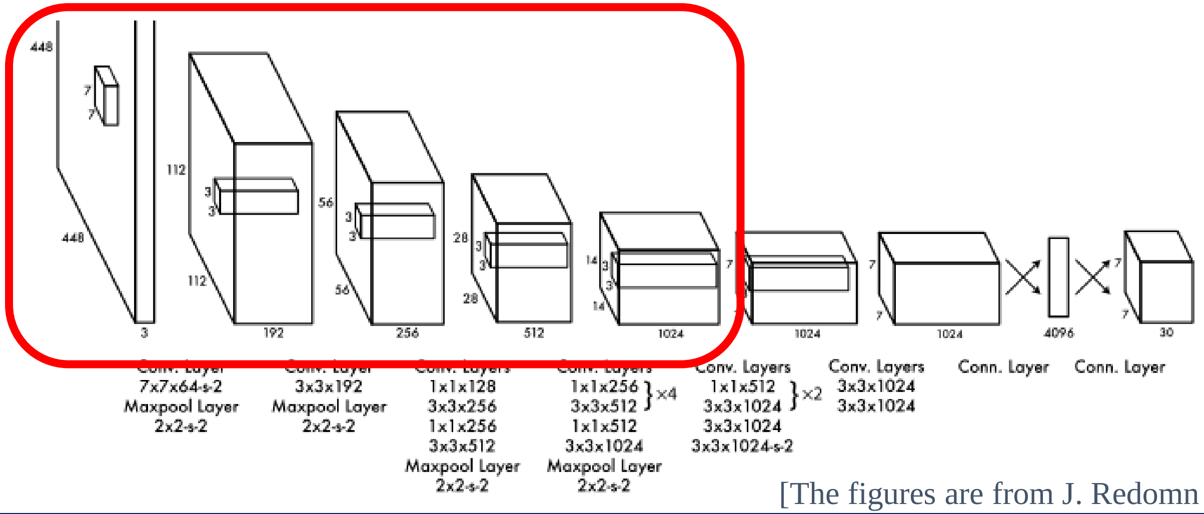
- **One CNN** to do all these things (24 conv layers and 2 fc layers)
  - Input is **an entire image**, not a single bounding box
  - The last layer does both box proposal and classification
  - The other layers extract a feature vector by analyzing the whole image



[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Training

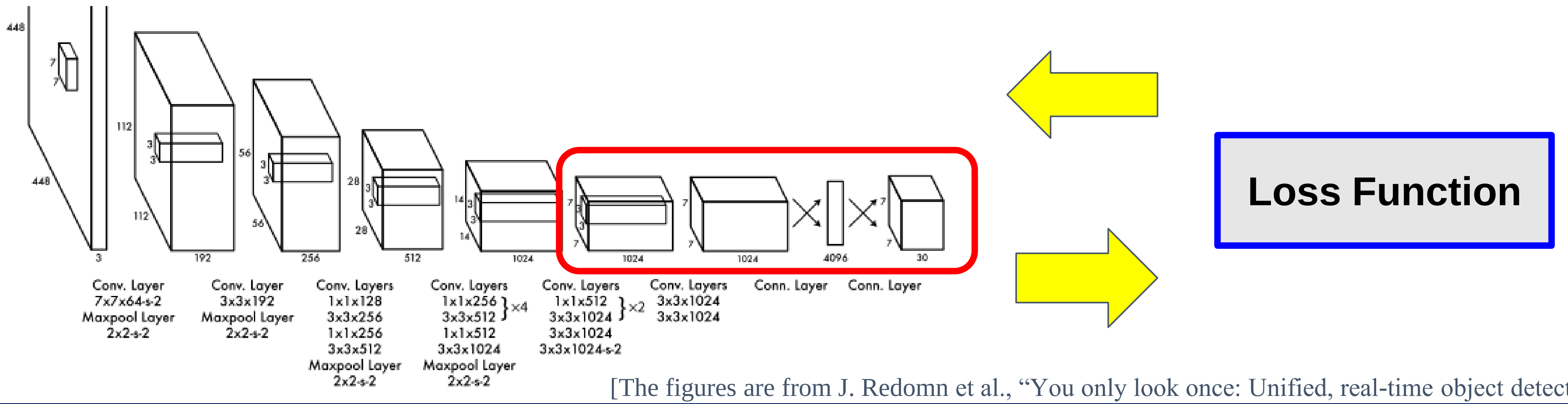
- Step 1) Pre-train a model for **classification**
  - First 20 conv layers followed by average-pooling layer and an fc layer (works well on ImageNet)



[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Training

- Step 2) Convert the model for **detection** (transfer learning)
  - Fix the pre-trained 20 conv layers and remove average pooling and fc layers
  - Add new 4 conv layers and 2 fc layers with random weights
  - Training the new 6 layers by using a **multi-task loss function**
    - **Only one box** is selected for each object, which have the highest IoU with the object's ground truth



[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR’16] – Loss Function

- **Remember!** We are training one model that does all the following tasks
  - Box proposal
  - Determine an object’s existence in the box (confidence)
  - Classify an object in the box if it exists

- The loss function is squared sum of (1) localization, (2) confidence, and (3) classification errors

Localization

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

Confidence

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2$$

Classification

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$$

[The figures are from J. Redomn et al., “You only look once: Unified, real-time object detection.”]

YoLo [CVPR'16] – Loss Function (Localization)

- Measure errors for center location and width/height

Center location

$$\boxed{\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]}$$

$$\boxed{+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]}$$

Width and height

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR’16] – Loss Function (Localization)

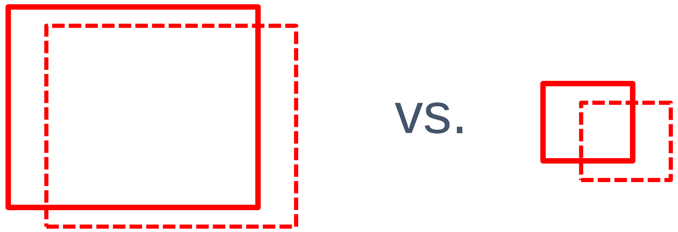
- Measure errors for center location and width/height
- If i-th cell has an object and its j-th box is “responsible” for the object, the loss function counts localization errors from the box

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \boxed{\mathbb{1}_{ij}^{\text{obj}}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \boxed{\mathbb{1}_{ij}^{\text{obj}}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

[The figures are from J. Redomn et al., “You only look once: Unified, real-time object detection.”]

YoLo [CVPR'16] – Loss Function (Localization)

- Measure errors for center location and width/height
- If i-th cell has an object and its j-th box is "responsible" for the object, the loss function counts localization errors from the box
- Why square root for width and height?
  - Need to boost errors for smaller boxes



$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \boxed{\left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]} \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR’16] – Loss Function (Localization)

- Measure errors for center location and width/height
- If i-th cell has an object and its j-th box is “responsible” for the object, the loss function counts localization errors from the box
- Why square root for width and height?
  - Need to boost errors for smaller boxes
- We should not equally weight localization error and classification error
  - The weight for localization error is **5 times higher** than that for classification error

$$\boxed{\lambda_{\text{coord}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$\boxed{+ \lambda_{\text{coord}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

[The figures are from J. Redomn et al., “You only look once: Unified, real-time object detection.”]

YoLo [CVPR'16] – Loss Function (Confidence)

- Measure errors for existence and absence
  - Confidence label = Pr(Object) x IoU(truth, pred)

$$\lambda_{\mathbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\mathbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

When there is something

$$\boxed{+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2}$$

When there is nothing

$$\boxed{+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2}$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Loss Function (Confidence)

- Measure errors for existence and absence
  - Confidence label = Pr(Object) x IoU(truth, pred)

- Different weight for the two terms
  - There are much more absence cases than existence cases, making the absence term larger than the existence term
  - Our goal is not to detect an absence case well though…
    - Right answer for 9 background images + Wrong answer for 1 object image = 90% accuracy??
  - <u>Halve the weight for the absence term</u>

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$\boxed{+ \lambda_{\text{noobj}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Loss Function (Classification)

- Measure for errors for every class

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$\boxed{+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2}$$

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Loss Function (Classification)

- Measure for errors for every class

- Only if a cell has an object!

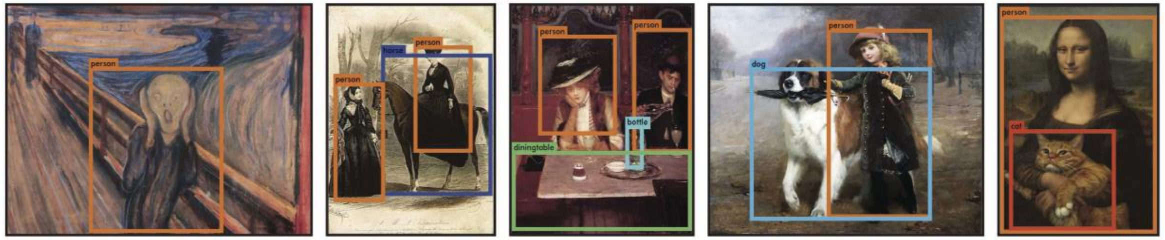
$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \boxed{\mathbb{1}_i^{\text{obj}}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Performance

- A bit less accurate but much **faster**! (40 s/image vs. 22 ms/image – 1800x)
- More accurate when applied to different image sets since YoLo is trained by using the whole image, not just boxes

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	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[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21



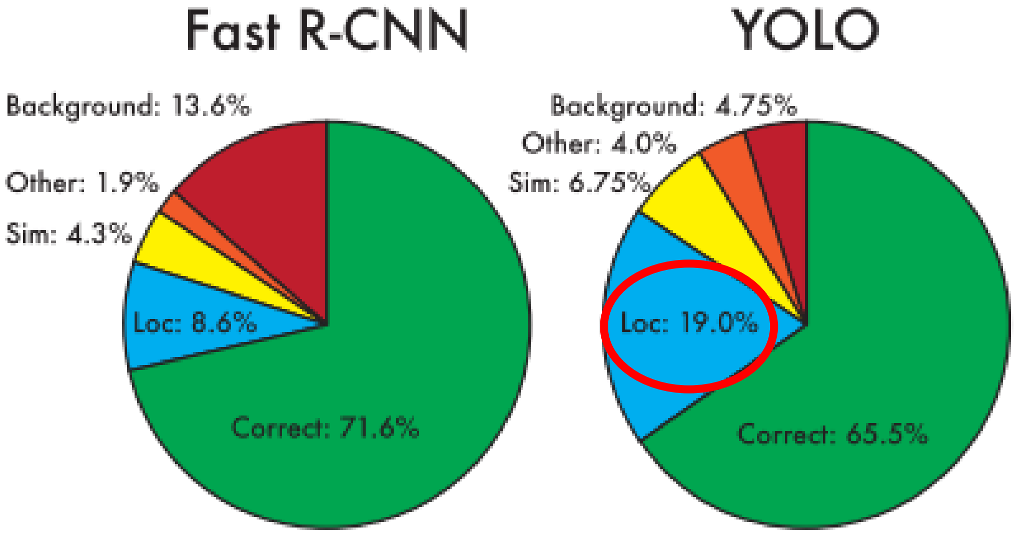
	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	**59.2**	**53.3**	**0.590**	**45**
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Problems

- Localization errors!
  - YoLo **learns** how to propose bounding boxes from **data**
    - It struggles to draw a bounding box for an object with **new or unusual** aspect ratios
  - Using square root for weight/height errors is not enough to resolve errors for small boxes

- Each cell can only have one class
  - It struggles to detect a group of small objects (e.g., flock of birds)



[The figures are from J. Redomn et al., "You only look once: Unified, real-time object detection."]

YoLo [CVPR'16] – Stepping Forward...

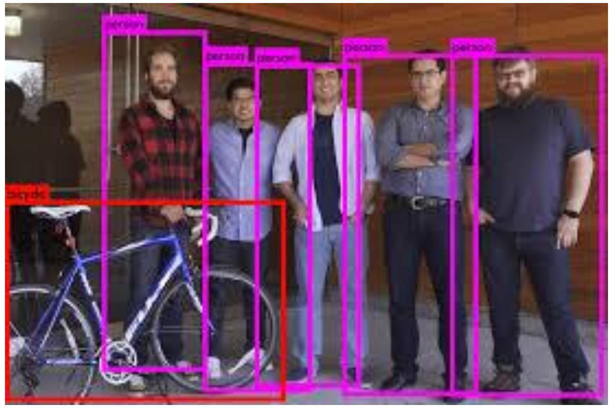
- YoLo v2 [CVPR'17] - Better, faster, and stronger
  - https://openaccess.thecvf.com/content_cvpr_2017/papers/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.pdf

- YoLo v3 [TechReport'18] - Minor fixes
  - https://arxiv.org/pdf/1804.02767.pdf;

YoLo [CVPR’16] – YoLo to Xnor.ai to Apple

- https://www.youtube.com/watch?time_continue=205&v=rov7T256z4s&feature=emb_logo)

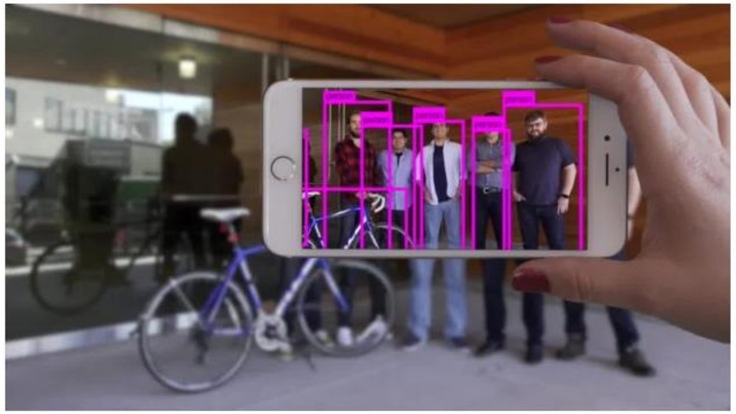
xnor.ai



Exclusive: Apple acquires Xnor.ai, edge AI spin-out from Paul Allen’s AI2, for price in $200M range

BY ALAN BOYLE, TAYLOR SOPER & TODD BISHOP on January 15, 2020 at 10:44 am

1 Comment | Share 520 | Tweet | Share | Reddit | Email



Become a GeekWire Member

GeekWire Newsletters

Subscribe to GeekWire's Space & Science weekly newsletter

Enter your email address | Subscribe

Send Us a Tip

Have a scoop that you'd like GeekWire to cover? Let us know.

*Can we detect **multiple objects** per cell while maintaining the concept of single-shot detector?*

SSD [ECCV'16] – Anchor Box

- We want to detect **multiple objects** per grid cell
- Each grid cell proposes a set of **predetermined** default anchor boxes with various aspect ratios (1,2,3,1/2,1/3), which are adjusted during the training procedure)

$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c1 \\ \vdots \\ cN \end{bmatrix}$$

YoLo
5B+C

$$\begin{bmatrix} b_x \\ b_y \\ b_h \\ b_w \\ c1 \\ \vdots \\ cN \\ b_x \\ b_y \\ b_h \\ b_w \\ c1 \\ \vdots \\ cN \end{bmatrix}$$

SSD (anchor box)
(C+4)xB

SSD [ECCV'16] – Resolution

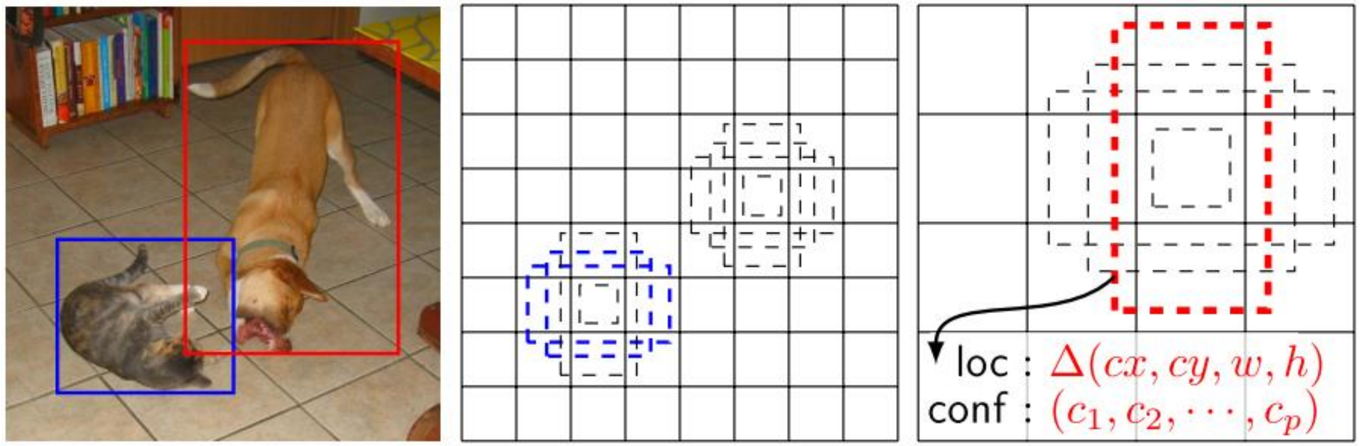
YoLo

(1) *One grid size*
(2) *Global bounding boxes*
(3) *Boxes learned from data*

SSD

(1) *Various grid sizes*
(2) *Local anchor boxes*
(3) *Adjust predetermined boxes*

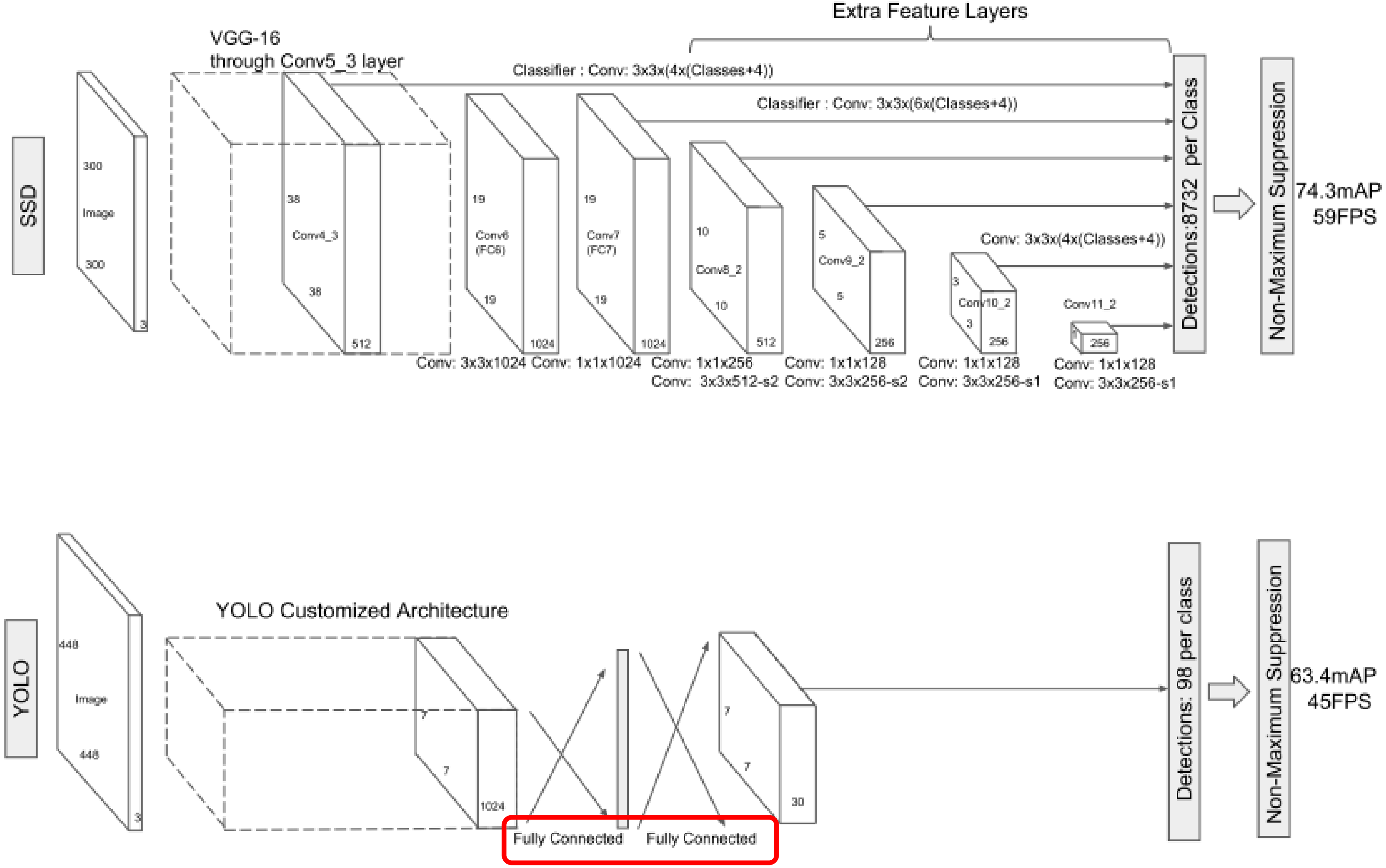
S × S grid on input   Bounding boxes + confidence



(a) Image with GT boxes   (b) 8×8 feature map   (c) 4×4 feature map

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

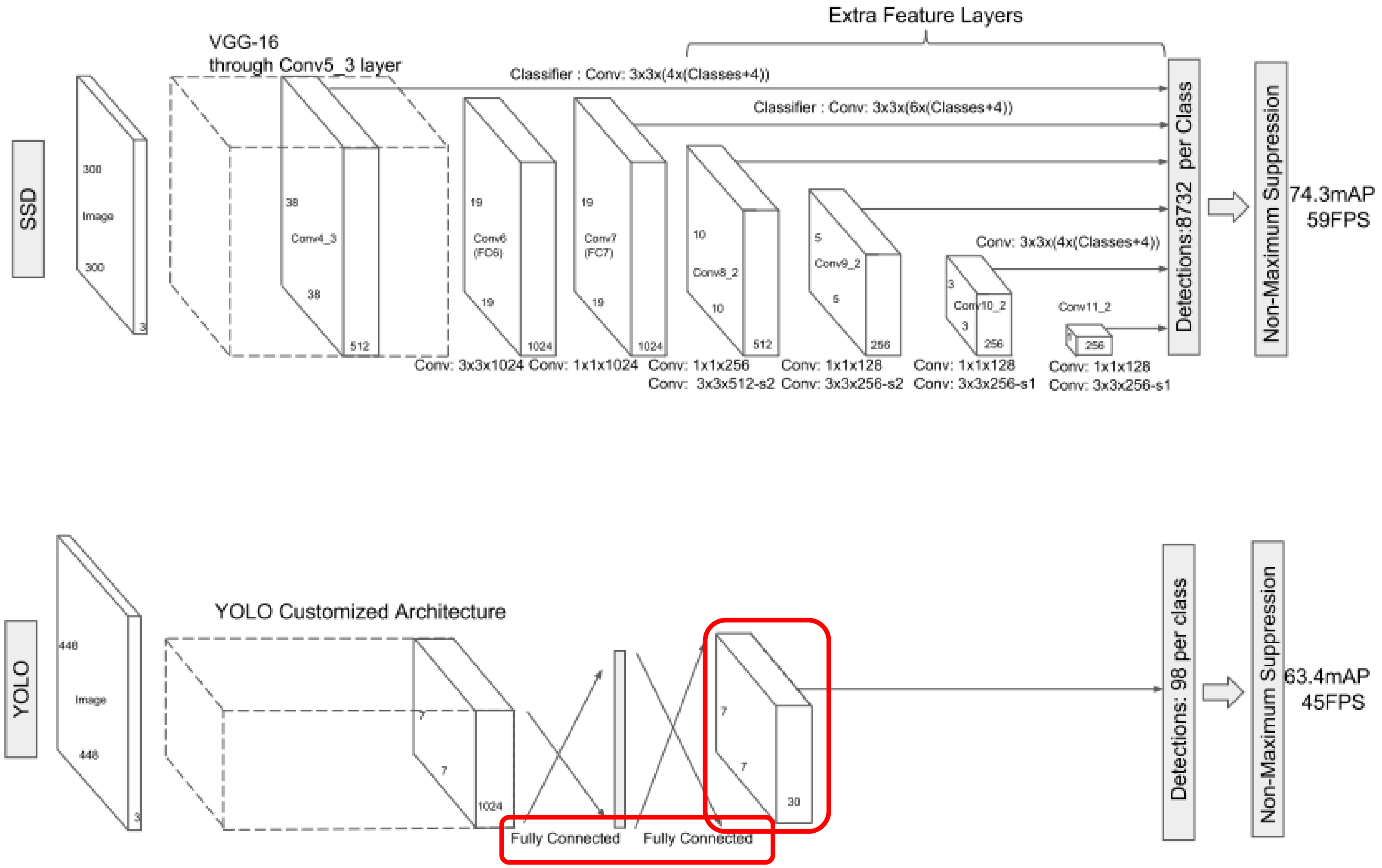
SSD [ECCV'16] – Architecture



***FC** for capturing global context*

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

SSD [ECCV’16] – Architecture

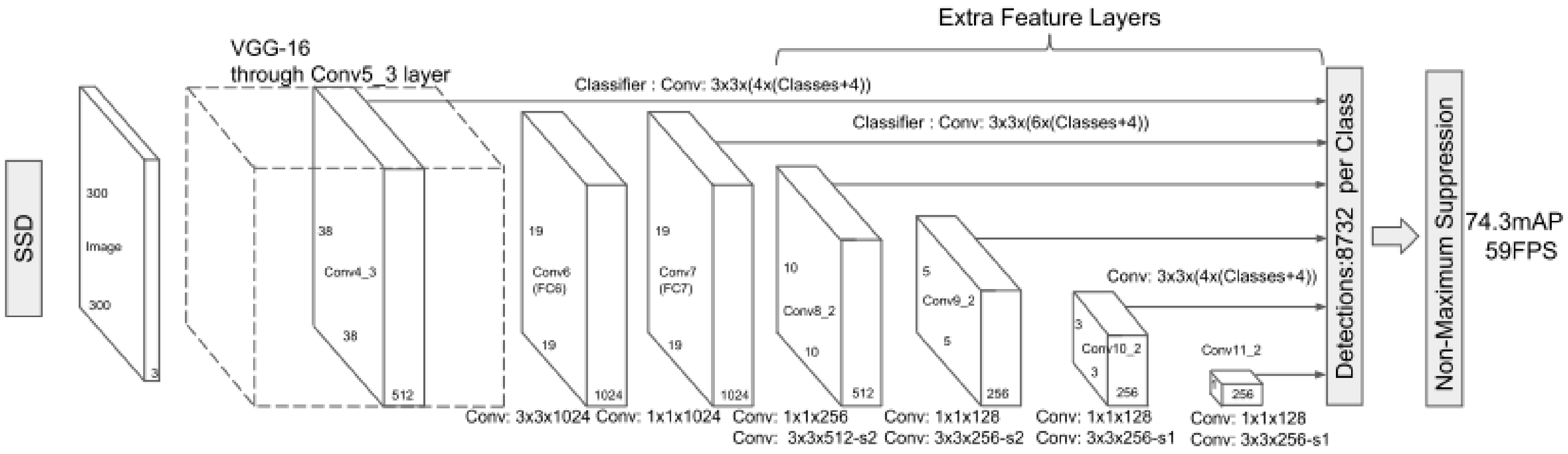


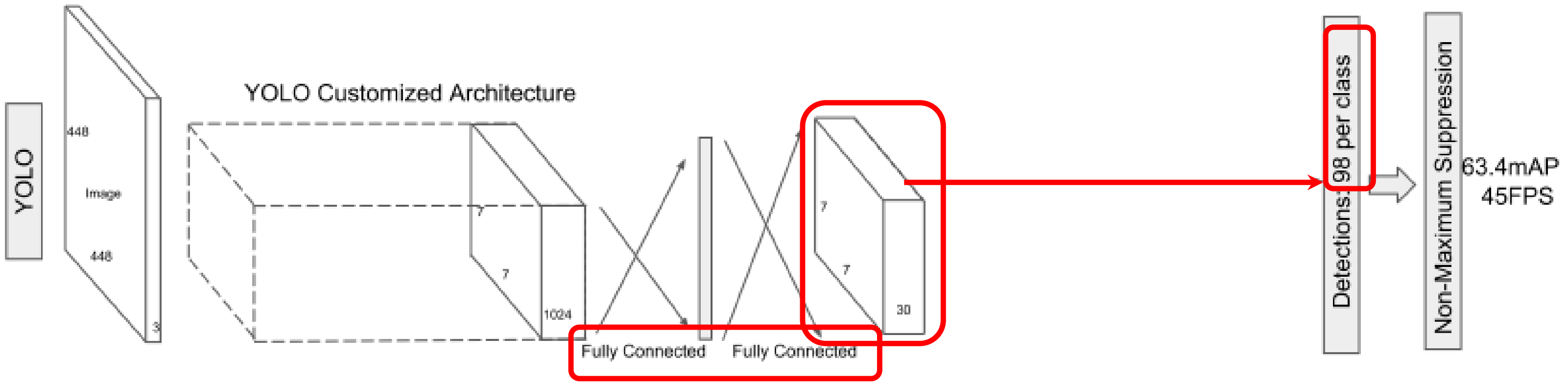
***FC** for capturing global context*

Lack of fine details

[The figures are from W. Liu et al., “SSD: Single shot multibox detector.”]

SSD [ECCV'16] – Architecture





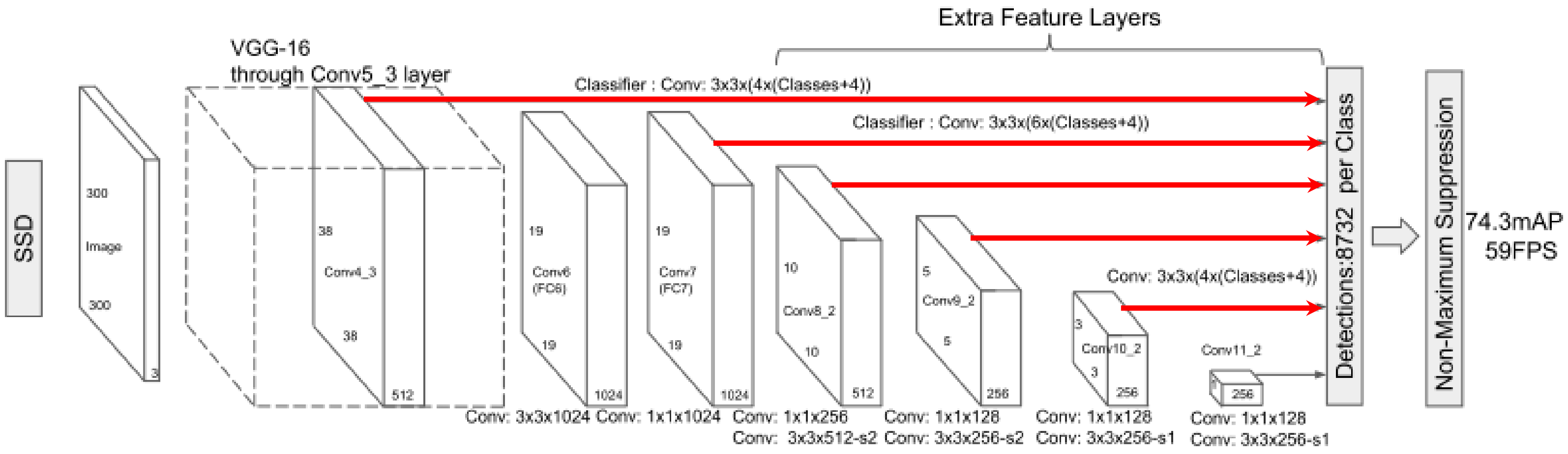
FC *for capturing global context*

Lack of fine details

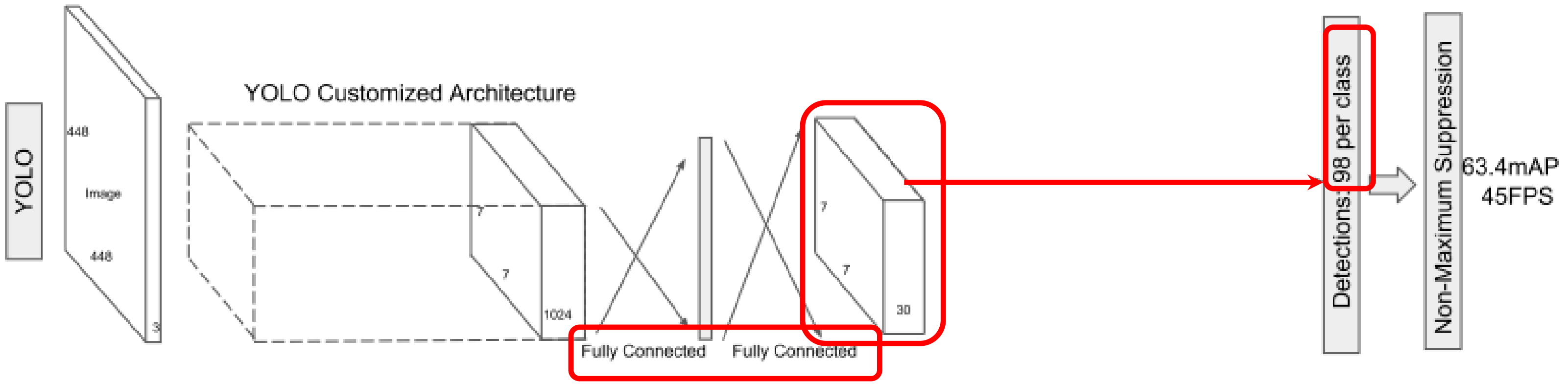
Only 98 bounding boxes

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

SSD [ECCV'16] – Architecture



No FC, but only **Conv**!
(local anchor boxes)



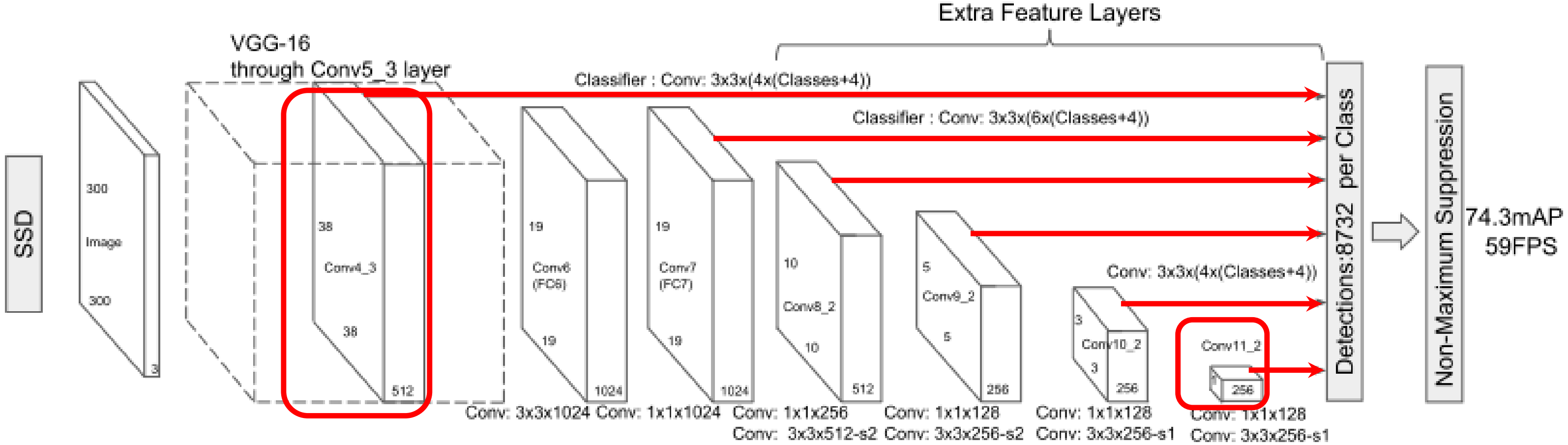
FC for capturing global context

Lack of fine details

Only 98
bounding boxes

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

SSD [ECCV'16] – Architecture

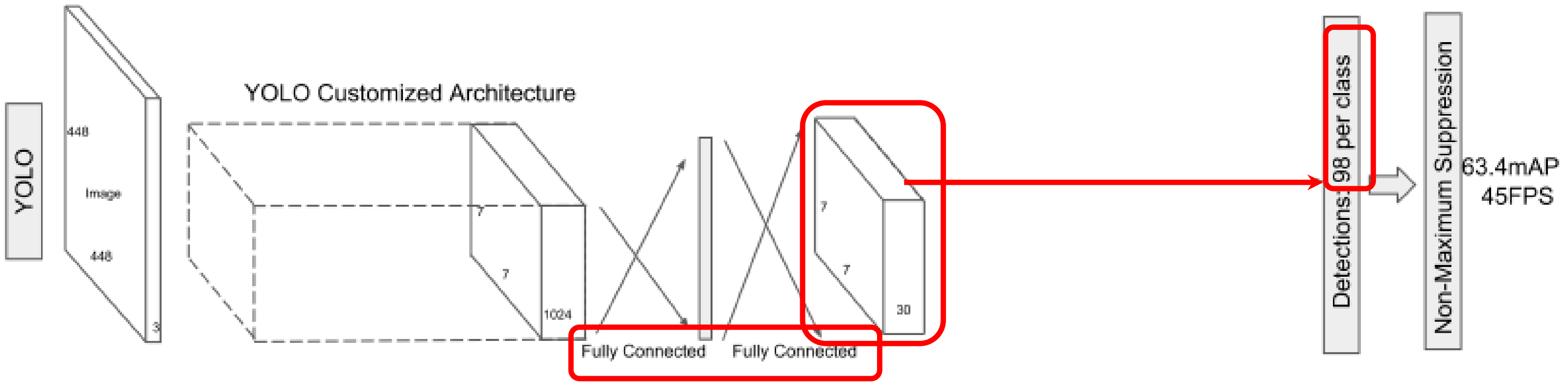


Fine details

Global context

*No FC, but only **Conv**!*
(local anchor boxes)

Both fine details and global context!



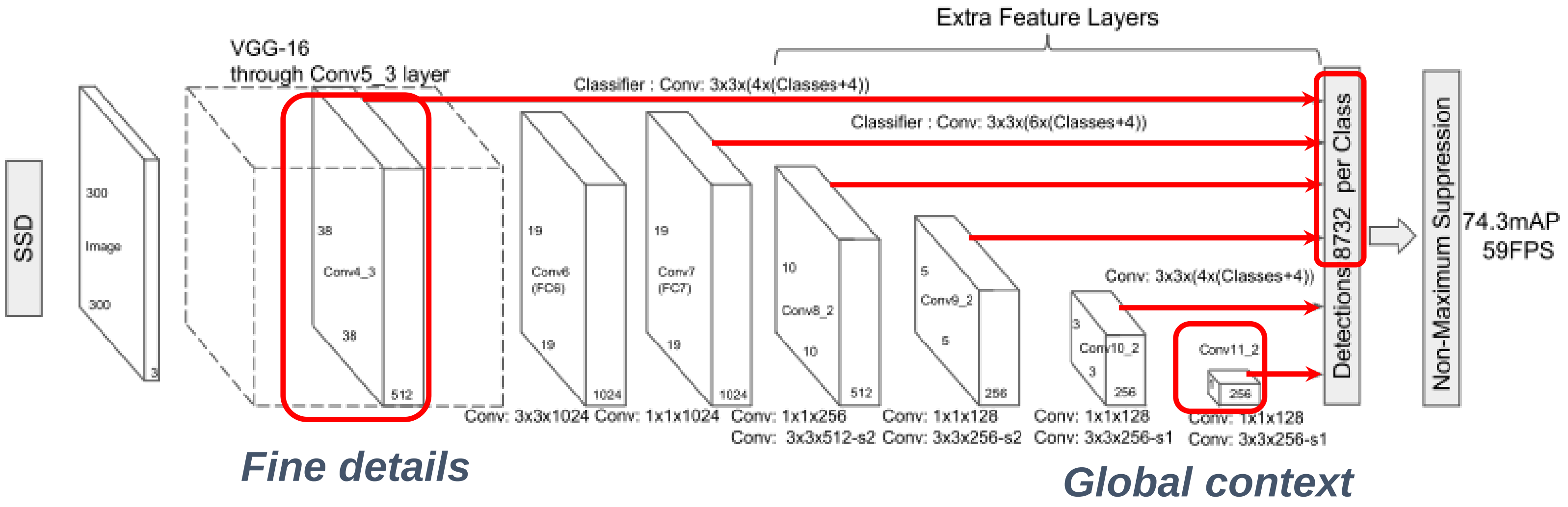
***FC** for capturing global context*

Lack of fine details

Only 98
bounding boxes

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

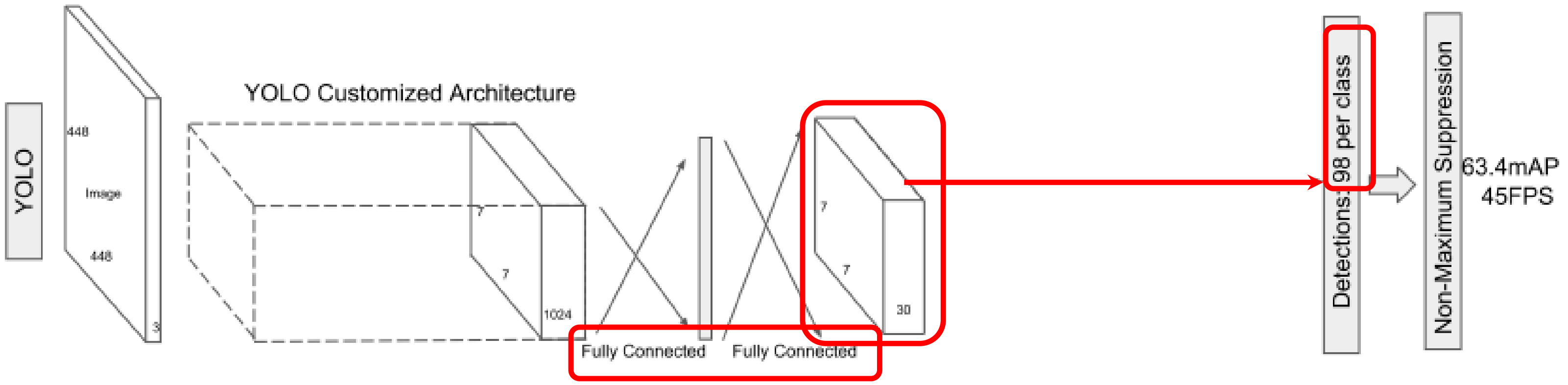
SSD [ECCV'16] – Architecture



*No FC, but only **Conv**!*
(local anchor boxes)

Both fine details and global context!

Much more anchor boxes (8732)!



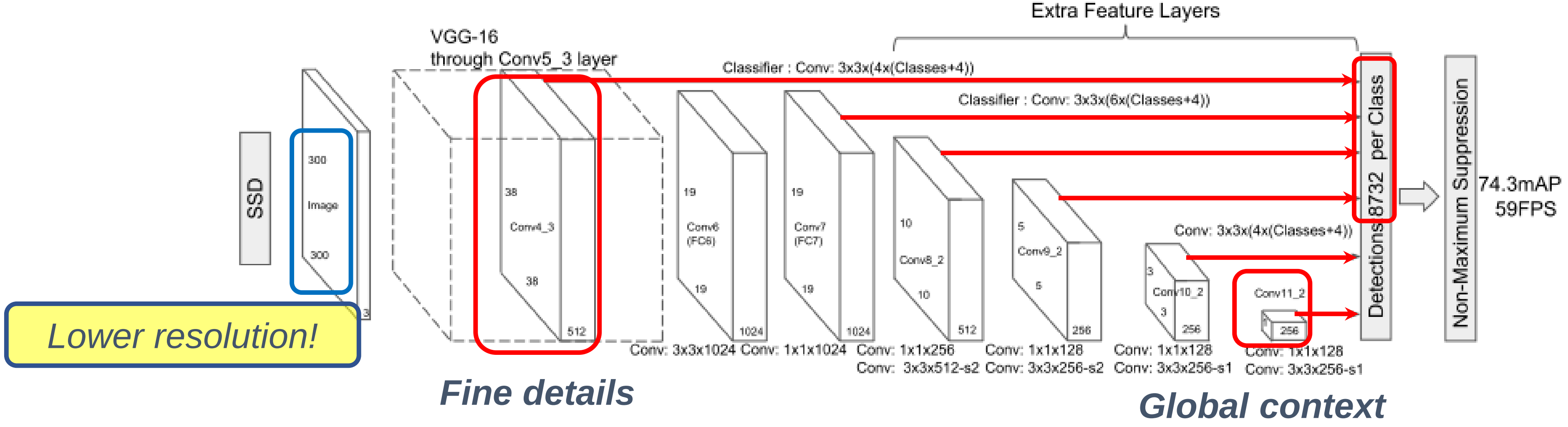
***FC** for capturing global context*

Lack of fine details

Only 98 bounding boxes

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

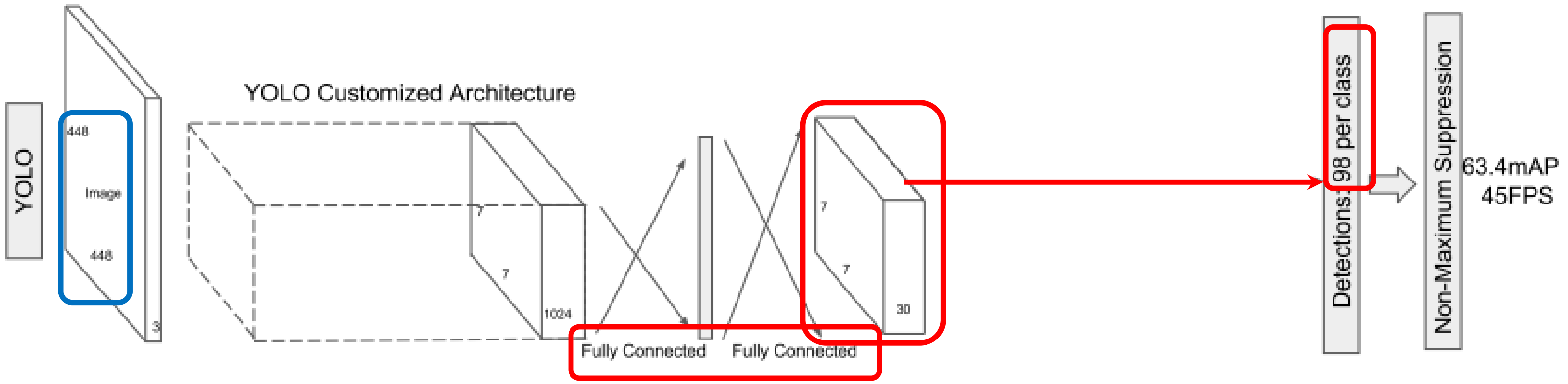
SSD [ECCV'16] – Architecture



No FC, but only **Conv**!
(local anchor boxes)

Both fine details and global context!

Much more anchor boxes (8732)!



FC *for capturing global context*

Lack of fine details

Only 98 bounding boxes

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

SSD [ECCV'16] – Loss Function

- Similar structure to YoLo's loss function, but **different** localization loss
  - For the center point

Ratio *compared to default box size*
(Larger error for a smaller default box)

$$L_{loc,center} = smooth_{L1}(\widehat{b_x} - \widehat{b_x^g}) + smooth_{L1}(\widehat{b_y} - \widehat{b_y^g})$$

$$\widehat{b_x} = \boxed{\frac{b_x - b_x^d}{b_w^d}} \qquad \widehat{b_x^g} = \frac{b_x^g - b_x^d}{b_w^d} \qquad \widehat{b_y} = \frac{b_y - b_y^d}{b_h^d} \qquad \widehat{b_x^g} = \frac{b_h^g - b_h^d}{b_h^d}$$

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

SSD [ECCV’16] – Loss Function

- Similar structure to YoLo’s loss function, but **different** localization loss
  - For the center point

Ratio *compared to default box size*
(Larger error for a smaller default box)

$$L_{loc,center} = smooth_{L1}(\widehat{b_x} - \widehat{b_x^g}) + smooth_{L1}(\widehat{b_y} - \widehat{b_y^g})$$

$$\widehat{b_x} = \boxed{\frac{b_x - b_x^d}{b_w^d}} \qquad \widehat{b_x^g} = \frac{b_x^g - b_x^d}{b_w^d} \qquad \widehat{b_y} = \frac{b_y - b_y^d}{b_h^d} \qquad \widehat{b_x^g} = \frac{b_h^g - b_h^d}{b_h^d}$$

  - For the size

***Log**: Larger error when a default box is larger than its corresponding ground truth*

$$L_{loc,size} = smooth_{L1}(\widehat{b_w} - \widehat{b_w^g}) + smooth_{L1}(\widehat{b_h} - \widehat{b_h^g})$$

$$\widehat{b_w} = \boxed{log\left(\frac{b_w}{b_w^d}\right)} \qquad \widehat{b_x^g} = log\left(\frac{b_w^g}{b_w^d}\right) \qquad \widehat{b_h} = log\left(\frac{b_h}{b_h^d}\right) \qquad \widehat{b_h^g} = log\left(\frac{b_h^g}{b_h^d}\right)$$

[The figures are from W. Liu et al., “SSD: Single shot multibox detector.”]

SSD [ECCV’16] – Loss Function

- Similar structure to YoLo’s loss function, but **different** localization loss
  - For the center point

$$L_{loc,center} = smooth_{L1}(\widehat{b_x} - \widehat{b_x^g}) + smooth_{L1}(\widehat{b_y} - \widehat{b_y^g})$$

$$\widehat{b_x} = \boxed{\frac{b_x - b_x^d}{b_w^d}} \qquad \widehat{b_x^g} = \frac{b_x^g - b_x^d}{b_w^d} \qquad \widehat{b_y} = \frac{b_y - b_y^d}{b_h^d} \qquad \widehat{b_x^g} = \frac{b_h^g - b_h^d}{b_h^d}$$

  - For the size

$$L_{loc,size} = smooth_{L1}(\widehat{b_w} - \widehat{b_w^g}) + smooth_{L1}(\widehat{b_h} - \widehat{b_h^g})$$

$$\widehat{b_w} = \boxed{log\left(\frac{b_w}{b_w^d}\right)} \qquad \widehat{b_x^g} = log\left(\frac{b_w^g}{b_w^d}\right) \qquad \widehat{b_h} = log\left(\frac{b_h}{b_h^d}\right) \qquad \widehat{b_h^g} = log\left(\frac{b_h^g}{b_h^d}\right)$$

  - Smooth function

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & if \ \ |x| < 1 \\ \boxed{|x| - 0.5, \quad otherwise} & \end{cases}$$

***Ratio** compared to default box size*
(Larger error for a smaller default box)

***Log**: Larger error when a default box is larger than its corresponding ground truth*

Mitigate gradient explosion due to outliers

[The figures are from W. Liu et al., “SSD: Single shot multibox detector.”]

SSD [ECCV'16] – Loss Function

- Similar structure to YoLo's loss function, but different localization loss
  - Complete loss for a pair of ground truth and its best anchor box

$$L \quad = L_{conf} + \alpha\left(L_{loc,center} + L_{loc,size}\right)$$

$$\alpha = 1 \ (instead\ of\ 5)$$

YoLo
Learn how to draw bounding boxes **from scratch**

SSD
Learn how to **adjust** (not too bad) default anchor boxes

[The figures are from W. Liu et al., "SSD: Single shot multibox detector."]

Stepping Forward …

- RetinaNet [ICCV’17]
  - Focal loss for dense object detection
- Pelee [NIPS’18]
  - Pelee: A real-time object detection system on mobile devices
- EfficientDet [CVPR’20]
  - EfficientDet: Scalable and Efficient Object Detection

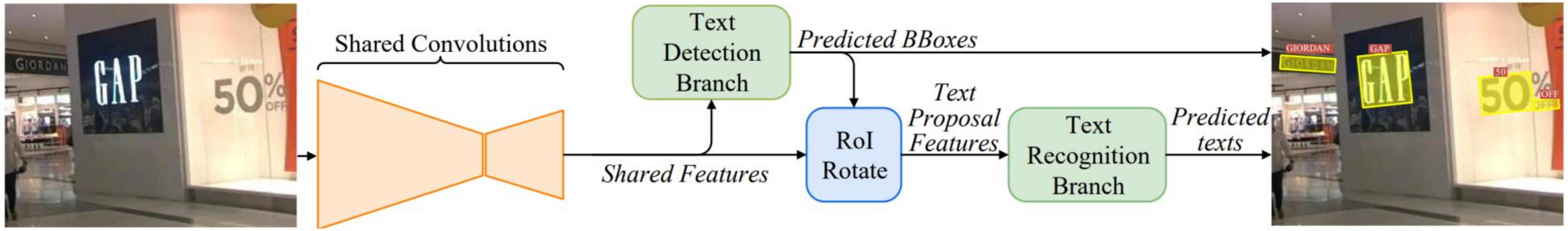
Summary

- Object detection = Classification and **localization** of multiple objects
  - Box proposal, IoU, NMS

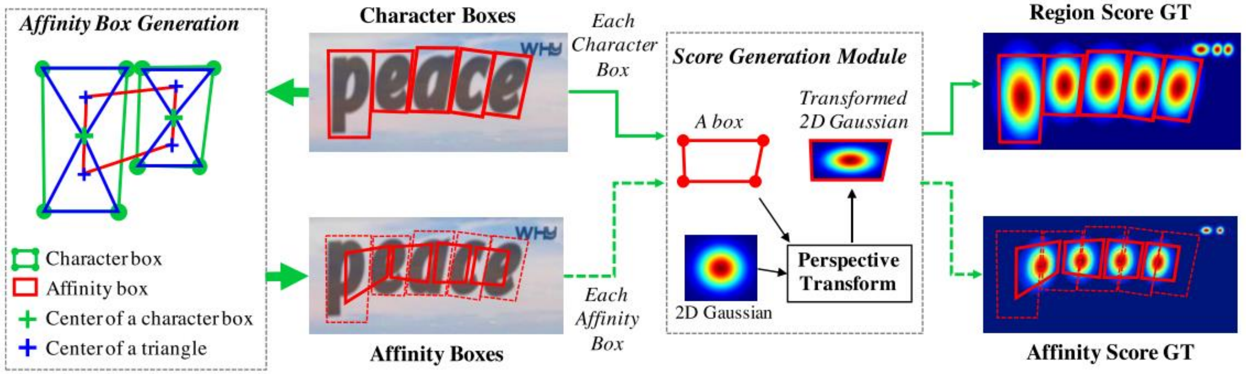
- Sliding windows
  - Intuitive but extremely slow (Run CNN for each of too many boxes)
- R-CNN
  - Run CNN for each of (only) 2000 boxes (selective search) but still slow
- SPPnet, Fast R-CNN, and Faster R-CNN
  - Much more advanced compared to R-CNN
- YoLo
  - Run CNN once and propose 98 boxes, fast but not accurate
- SSD
  - Run CNN once without FC layers and propose 8732 boxes, fast and accurate

Object + Text!

- FOTS [CVPR'18]
  - Fots: Fast oriented text spotting with a unified network



- CRAFT [CVPR'19]
  - Character region awareness for text detection



Thanks!