

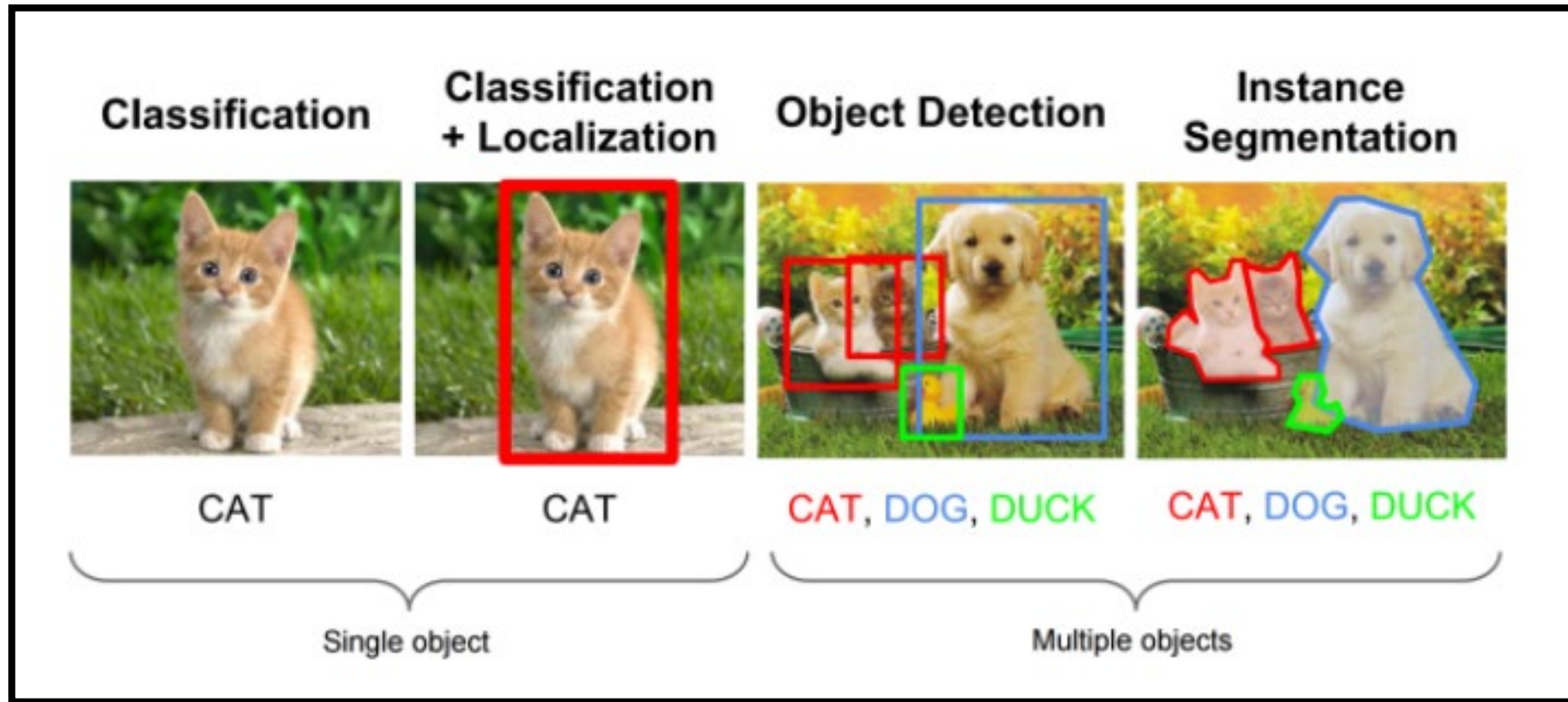


› AUTOSYS – DEEP LEARNING

SSD – Single Shot Detector

WHAT IS THE DIFFERENCE BETWEEN:

CLASSIFICATION | *LOCALIZATION* | *DETECTION* | *SEGMENTATION* ?



[SOURCE] https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/object_localization_and_detection.html

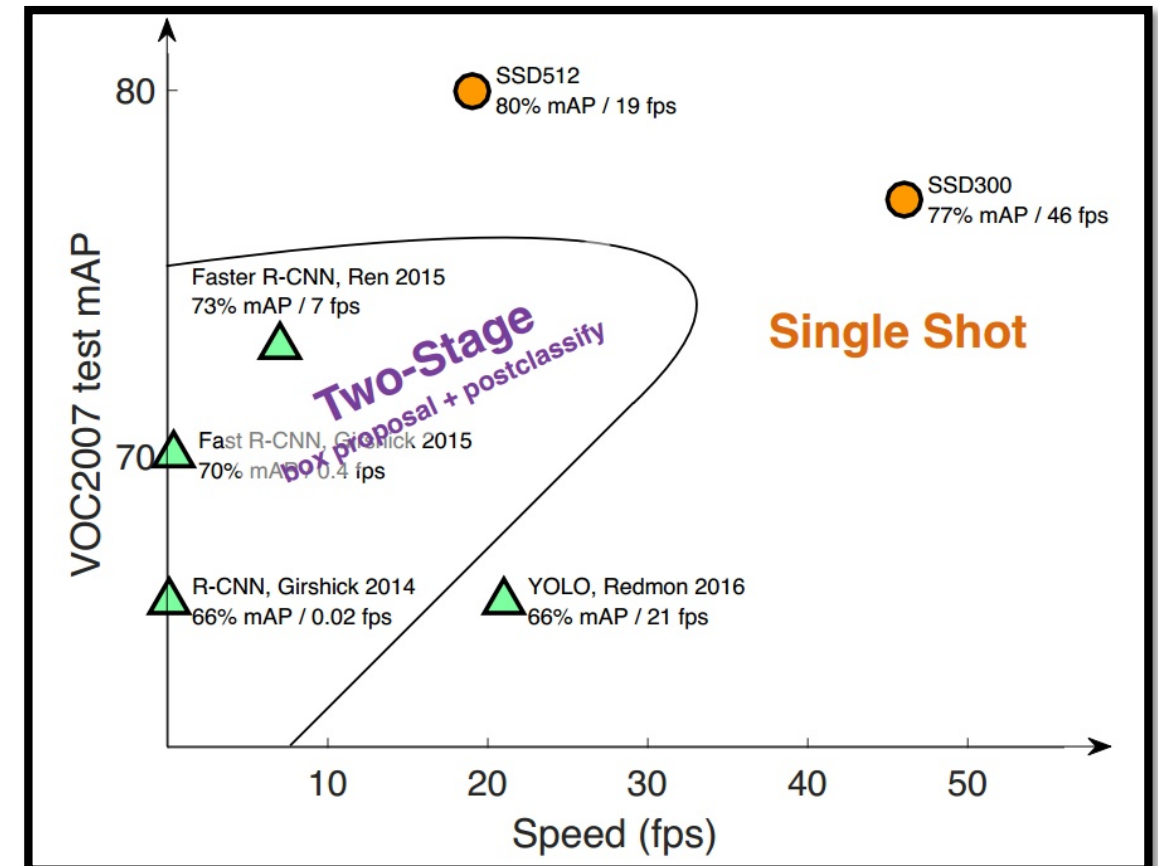
DIFFERENT APPROACHES IN THE LAST FEW YEARS ...

SINGLE SHOT DETECTOR

One Forward pass for both “Box Proposal” and “Classification”.

TWO STAGE DETECTOR

Two forward passes for both “Box Proposal” and “Classification”



[SOURCE] <https://zhuanlan.zhihu.com/p/30478644>

SSD – HIGH LEVEL VIEW



[SOURCE] https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

HOW TO LOCALIZE OBJECTS WITH REGRESSION?

- With regression we can adjust/update numbers rather than finding the correct class.
- To define a Bounding Box fully we need 4 Numbers:

X, Y, Width, Height


- We need a function to calculate the loss between predicted bounding box and ground truth.

Smooth L1 / Jaccard Index

- Now we can update the weights to get closer to the ground truth bounding box.

UNDERSTANDING IOU

INTERSECTION OVER UNION (JACCARD INDEX)




GROUND TRUTH BOX

AREA OF OVERLAP

ANCHOR BOX

$Area\ GTB = 5\ cm \cdot 5\ cm = 25$
 $Area\ AB = 10\ cm \cdot 5\ cm = 50$
 $Area\ of\ Overlap = 12.5$
 $Area\ of\ Union = 62.5$
 $IoU = \frac{12.5}{62.5} = 0.2 \rightarrow \text{Negative}$

$Area\ GTB = 5\ cm \cdot 5\ cm = 25$
 $Area\ AB = 5\ cm \cdot 5\ cm = 25$
 $Area\ of\ Overlap = 20$
 $Area\ of\ Union = 30$
 $IoU = \frac{20}{30} = 0.66 \rightarrow \text{Positive}$

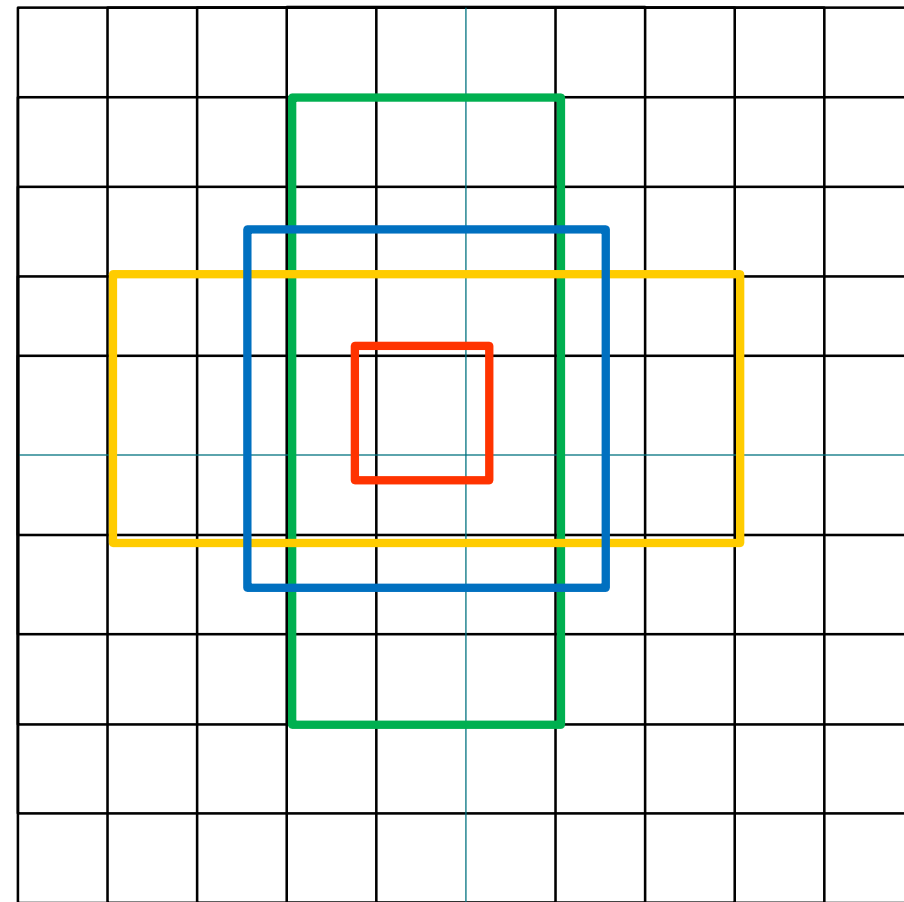
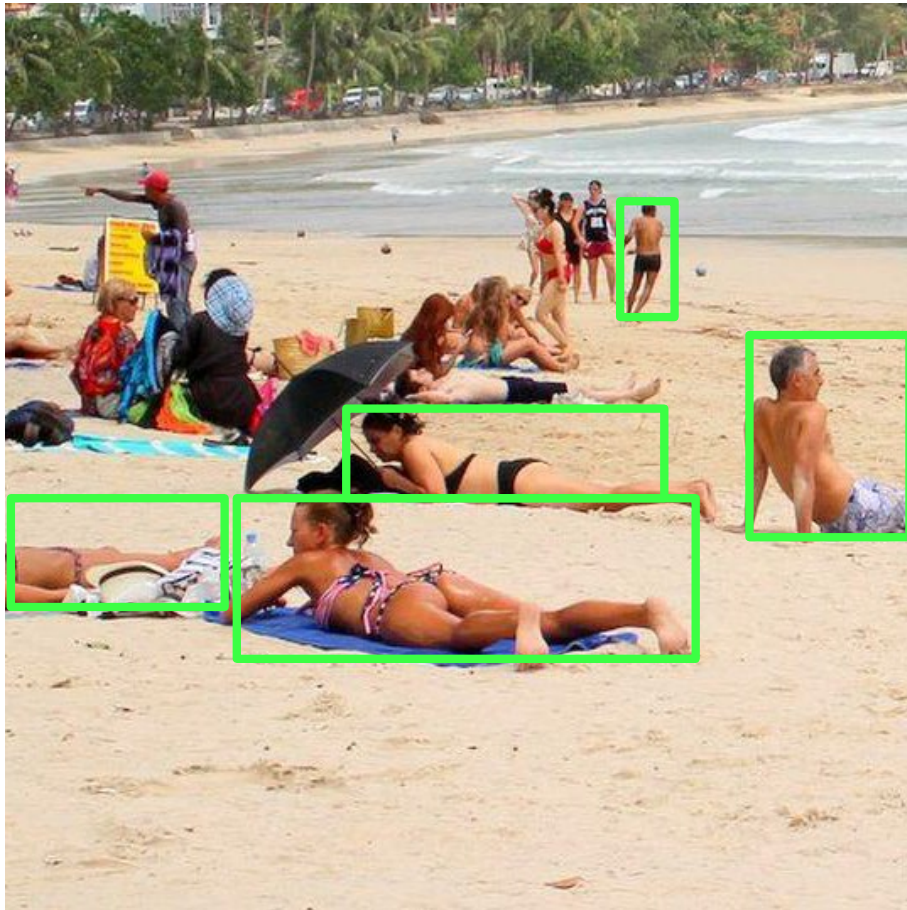
$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$


Positive IoU Threshold
[0.5, 1.0]
Ignored by the network
(0.2, 0.5)
Negative IoU Threshold
[0, 0.2]

USING DEFAULT BOXES (ALSO KNOW AS PRIORS OR ANCHOR BOXES)

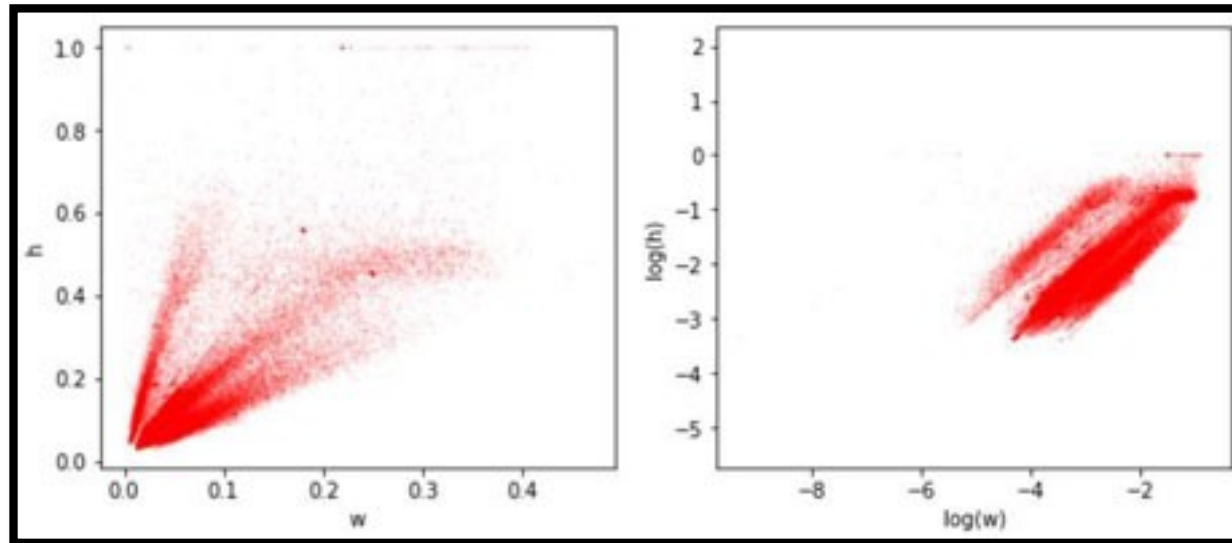
- **What is it?** Default Boxes are a collection of boxes overlaid on the image at different spatial **locations**, **scales** and **aspect ratios** that act as reference points on the ground truth bounding boxes
- **Why do we need it?** We could start with random predictions and use gradient descent to optimize the model. However, during the initial training, the model may fight with each other to determine what shapes to be optimized for which predictions. → No Convergence of the model, slow training, ...
- A model is then trained to make two predictions for each default box:
 - A discrete class prediction for each default box
 - A continuous prediction of an offset by which the anchor needs to be shifted to fit the ground-truth bounding box

WHAT KIND OF DEFAULT BOXES ARE SENSIBLE?



ARE PRE-DEFINED ASPECT RATIOS LIMITING?

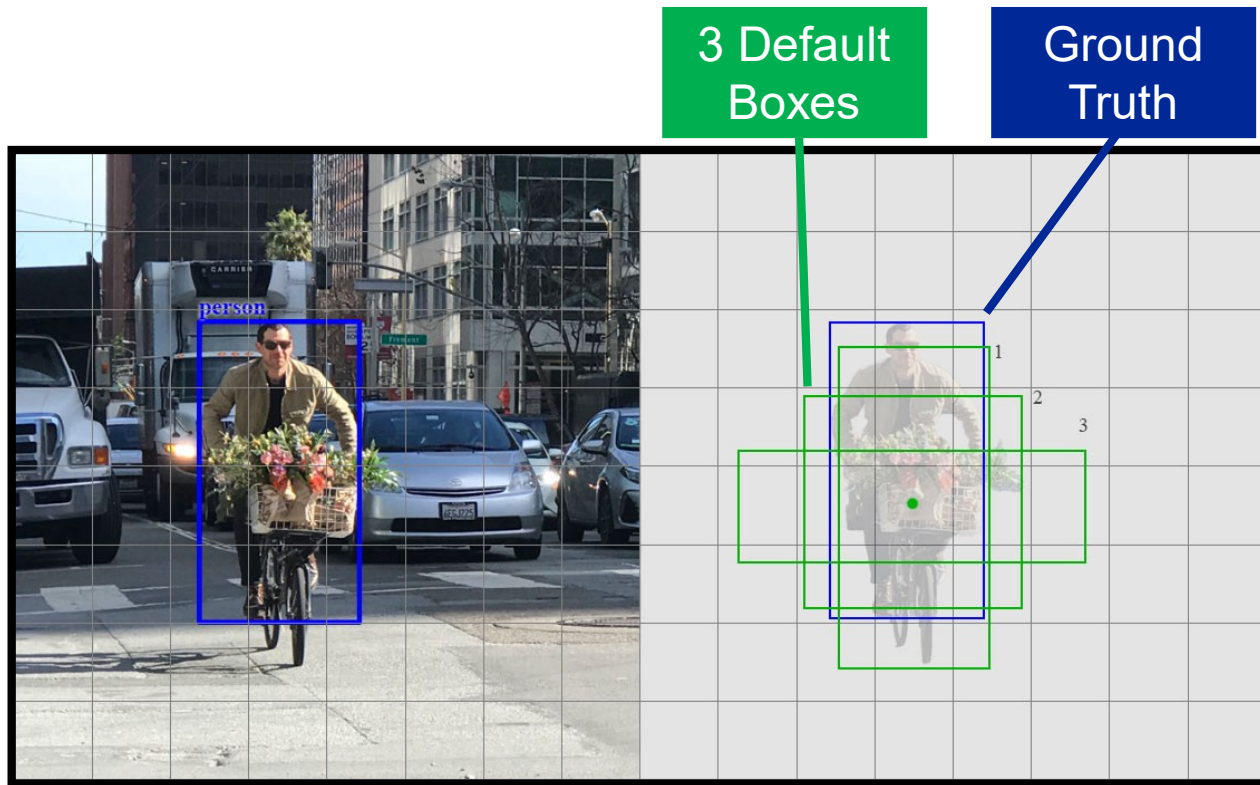
No, because **boundary boxes do not have arbitrary shape and size.**



The graph above shows the width and height distributions for the bounding boxes of the KITTI Dataset. Those distributions are highly clustered.

[SOURCE] https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

MATCHING STRATEGY

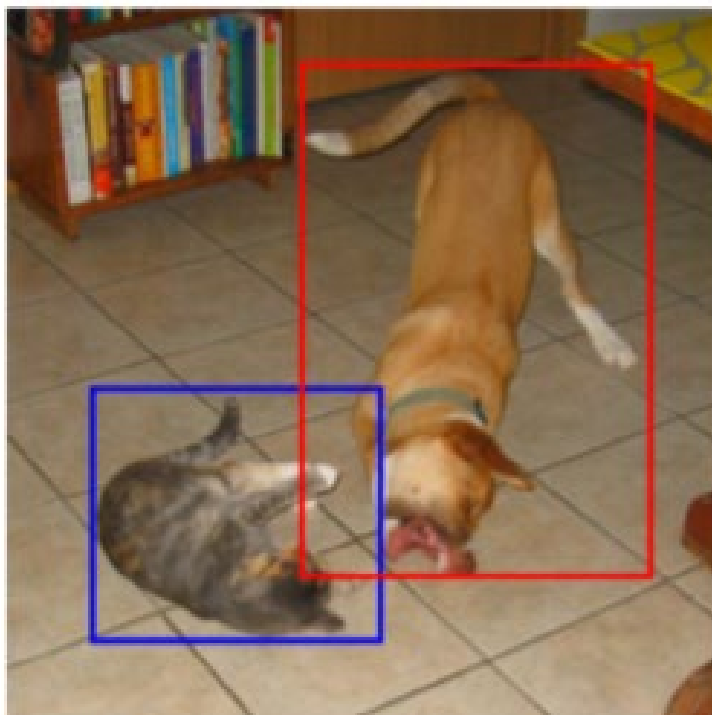


- The cost for the boundary mismatch is only calculated when the IoU is greater than 0.5
- In this case we simplify the case and only have 3 *Default Boxes*.
- Default Box 3 is discarded because the IoU value is lower than 0.5
- Default Box 1 and 2 have a IoU value greater than 0.5 → Calculate the localization cost of the corresponding predicted bounding box → Take the best.

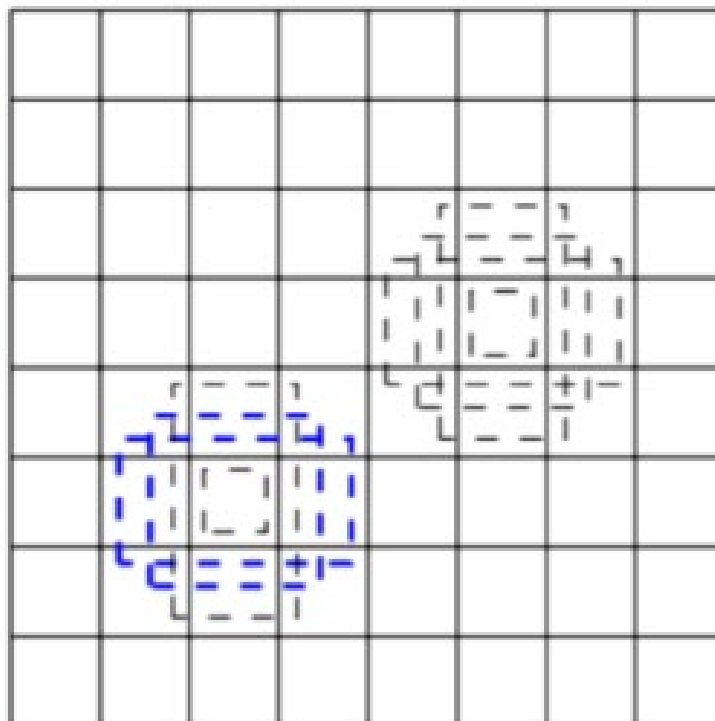
THIS MATCHING STRATEGY ENCOURAGES EACH PREDICTION TO PREDICT SHAPES CLOSER TO THE CORRESPONDING DEFAULT BOX. THEREFORE OUR PREDICTIONS ARE MORE DIVERSE AND MORE STABLE IN THE TRAINING.

[SOURCE] https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

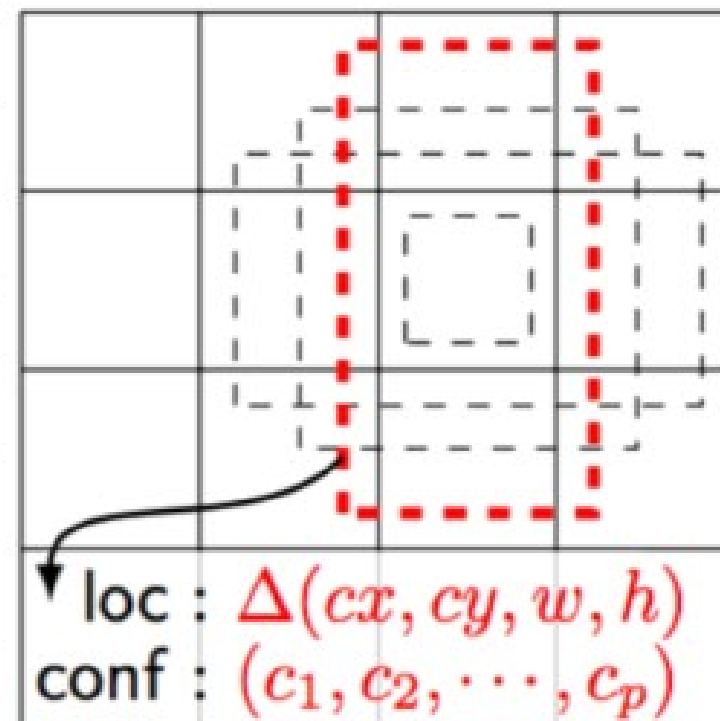
HOW TO BE SCALE INVARIANT?



(a) Image with GT boxes



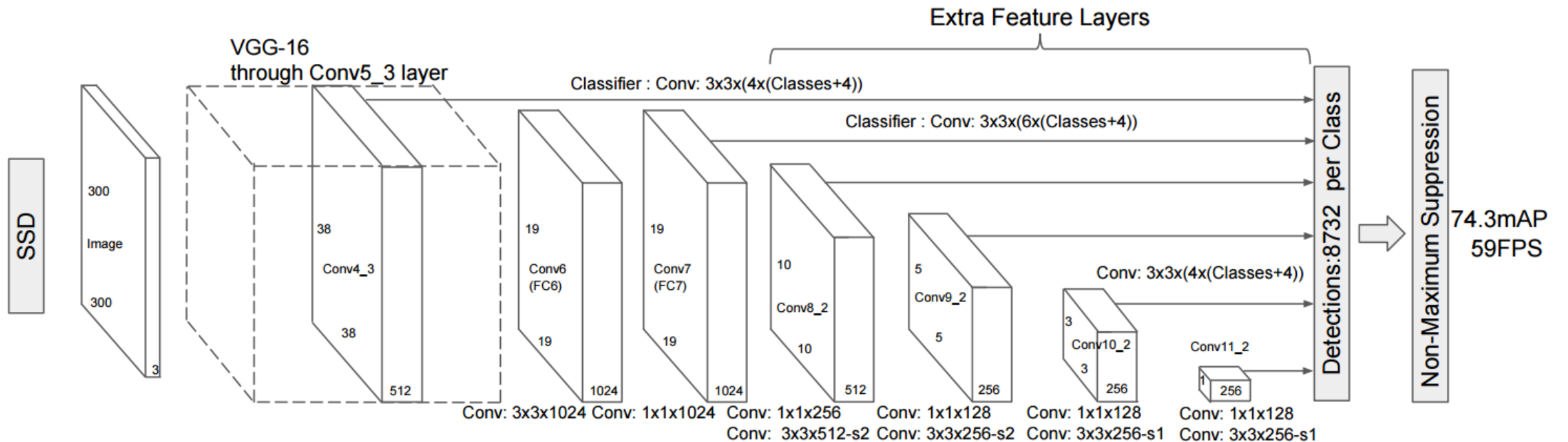
(b) 8×8 feature map



(c) 4×4 feature map

[SOURCE] <https://arxiv.org/pdf/1512.02325.pdf>

HOW TO BE SCALE INVARIANT?



[SOURCE] <https://arxiv.org/pdf/1512.02325.pdf>

LOSS FUNCTION

The **localization loss** is the mismatch between the ground truth box and the predicted boundary box. SSD only penalizes predictions from positive matches. We want the predictions from the positive matches to get closer to the ground truth. Negative matches can be ignored.

The localization loss between the predicted box l and the ground truth box g is defined as the smooth L1 loss with cx, cy as the offset to the default bounding box d of width w and height h .

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

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

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

$$x_{ij}^p = \begin{cases} 1 & \text{if } IoU > 0.5 \text{ between default box } i \text{ and ground true box } j \text{ on class } p \\ 0 & \text{otherwise} \end{cases}$$

[SOURCE] <https://arxiv.org/pdf/1512.02325.pdf>

LOSS FUNCTION

The **confidence loss** is the loss in making a class prediction. For every positive match prediction, we penalize the loss according to the confidence score of the corresponding class. For negative match predictions, we penalize the loss according to the confidence score of the class “0”: class “0” classifies no object is detected.

It is calculated as the softmax loss over multiple classes confidences c (class score).

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

where N is the number of matched default boxes.

The **final loss function** looks like that:

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

[SOURCE] <https://arxiv.org/pdf/1512.02325.pdf>

HARD NEGATIVE MINING

PROBLEM

- We make far more predictions than the number of objects present in the picture.
- This means we have much more negative matches than positive matches.
- This creates an imbalance → Bad for Training

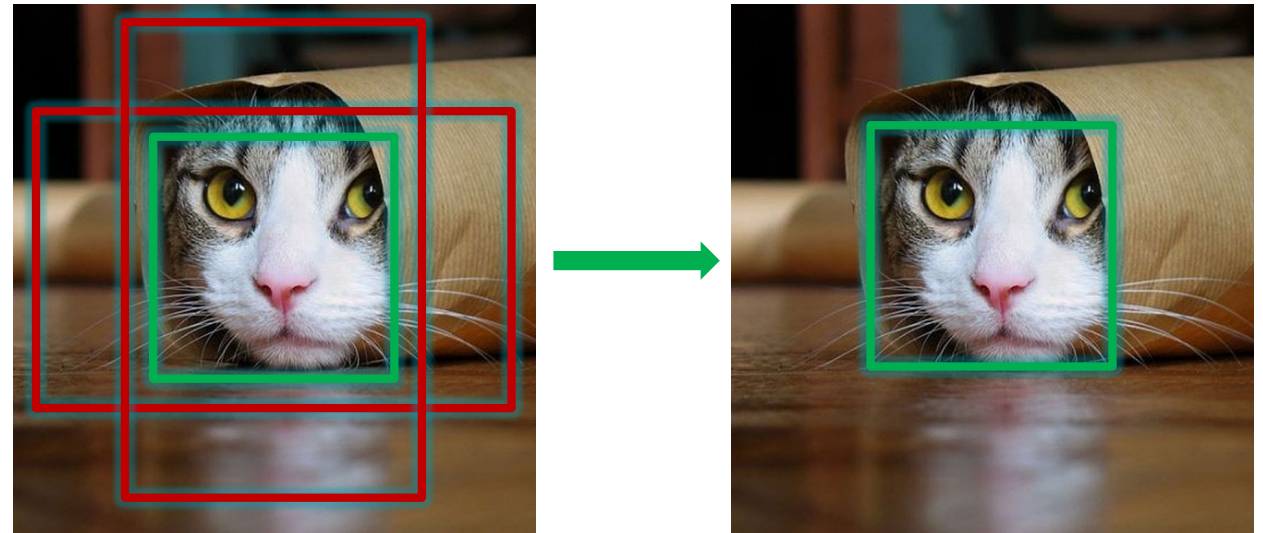
SOLUTION – HARD NEGATIVE MINING

- Sort negative matches by their confidence loss
- Pick the negative matches with the top loss
- Make sure the ratio between negative and positive matches is not bigger than 3:1

NON-MAXIMUM SUPPRESSION

SSD uses **non-maximum suppression** to remove duplicate predictions pointing to the same object.

- SSD sorts the predictions by the confidence scores.
- Start from the top confidence prediction, SSD evaluates whether any previously predicted boundary boxes have an IoU higher than 0.45 with the current prediction for the same class.
- If found, the current prediction will be ignored.
- Deeper Understanding: <https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c>



A QUICK WORD ON *DATA AUGMENTATION*

Data augmentation is **important** in improving accuracy. Augment data with flipping, cropping and color distortion.

data augmentation	SSD300		
horizontal flip	✓	✓	✓
random crop & color distortion		✓	✓
random expansion			✓
VOC2007 test mAP	65.5	74.3	77.2



[SOURCE] https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

A QUICK WORD ON *INFERENCE TIME*

SSD300 makes many predictions (8732) for a better coverage of location, scale and aspect ratios, more than many other detection methods.

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

[SOURCE] https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

TIPPS & TRICKS

- Accuracy increases with the number of default boundary boxes at the cost of speed.
- Adjust aspect ratios of your default bounding boxes in accordance to your dataset.
- If you have smaller objects to detect, it is necessary to increase resolution, as well at the cost of speed

[SOURCE] https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06