Boulder

Computer Vision; Object Detection; One Stage Detectors

YouTube Playlist

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

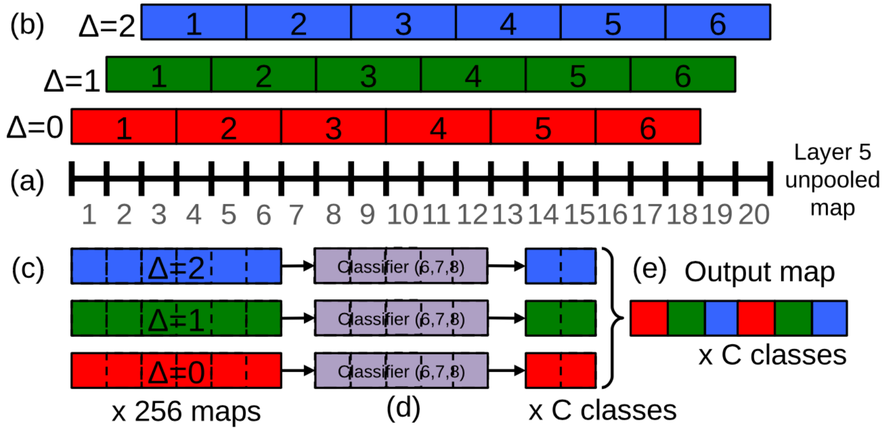
OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks

YouTube Video

Layer	1	2	3	4	5	6	7	8	Output 9
Stage	conv + max	conv + max	conv	conv	conv	conv + max	full	full	full
# channels	96	256	512	512	1024	1024	4096	4096	1000
Filter size	7x7	7x7	3x3	3x3	3x3	3x3	-	-	-
Conv. stride	2x2	1x1	1x1	1x1	1x1	1x1	-	-	-
Pooling size	3x3	2x2	-	-	-	3x3	-	-	-
Pooling stride	3x3	2x2	-	-	-	3x3	-	-	-
Zero-Padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input size	221x221	36x36	15x15	15x15	15x15	15x15	5x5	1x1	1x1

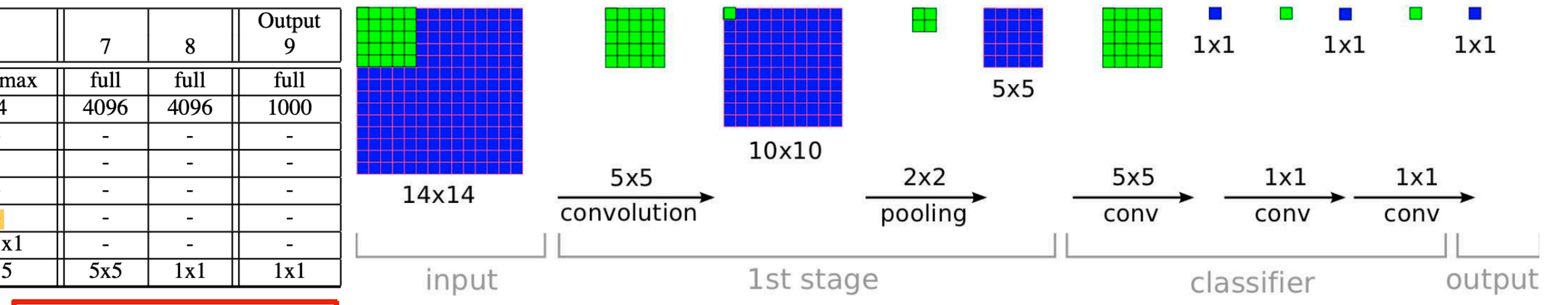
$36 = 2 \cdot 3 \cdot 2 \cdot 3 \rightarrow$ total subsampling ratio in the network

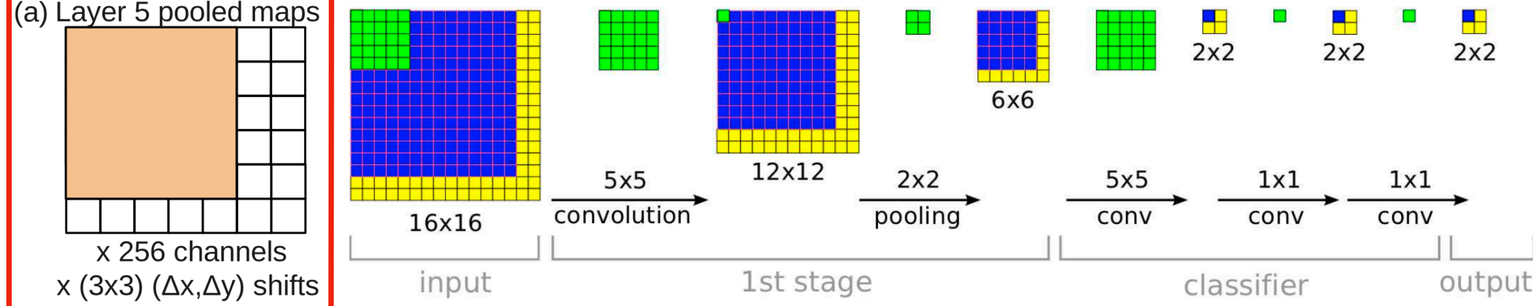
Scale	Input size	Layer 5 pre-pool	Layer 5 post-pool	Classifier map (pre-reshape)	Classifier map size
1	245x245	17x17	(5x5)x(3x3)	(1x1)x(3x3)xC	3x3xC
2	281x317	20x23	(6x7)x(3x3)	(2x3)x(3x3)xC	6x9xC
3	317x389	23x29	(7x9)x(3x3)	(3x5)x(3x3)xC	9x15xC
4	389x461	29x35	(9x11)x(3x3)	(5x7)x(3x3)xC	15x21xC
5	425x497	32x35	(10x11)x(3x3)	(6x7)x(3x3)xC	18x24xC
6	461x569	35x44	(11x14)x(3x3)	(7x10)x(3x3)xC	21x30xC



$245 = 221 + 2 \cdot 12 \rightarrow$ input size

Yields a total subsampling ratio of $12 = 36/3$ instead of 36





Greedy Merge Strategy

$C_s \rightarrow$ set of classes in the top k for each scale $s = 1, \ldots, 6$

$B_s \rightarrow$ set of bounding boxes predicted by the regressor network for each class in C_s

Assign $B \leftarrow \bigcup_s B_s$

Repeat merging until done:

$(b_1^*, b_2^*) = \text{argmin}_{b_1 \neq b_2 \in B} \texttt{match\_score}(\texttt{b}_1, \texttt{b}_2)$

If $\texttt{match\_score}(b_1^*, b_2^*) > t$, stop.

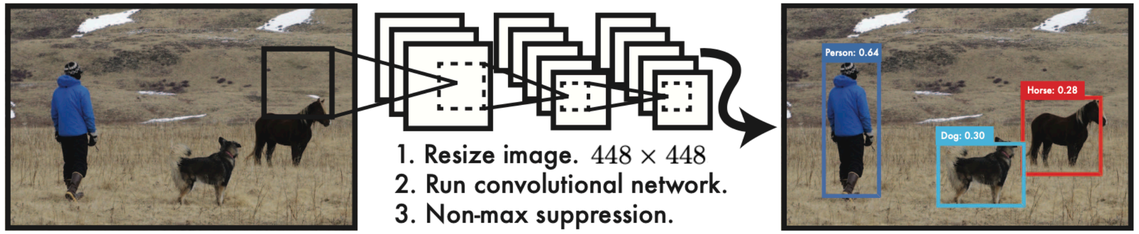
Otherwise, set $B \leftarrow B \backslash \{b_1^*, b_2^*\} \cup \texttt{box\_merge}(b_1^*, b_2^*)$

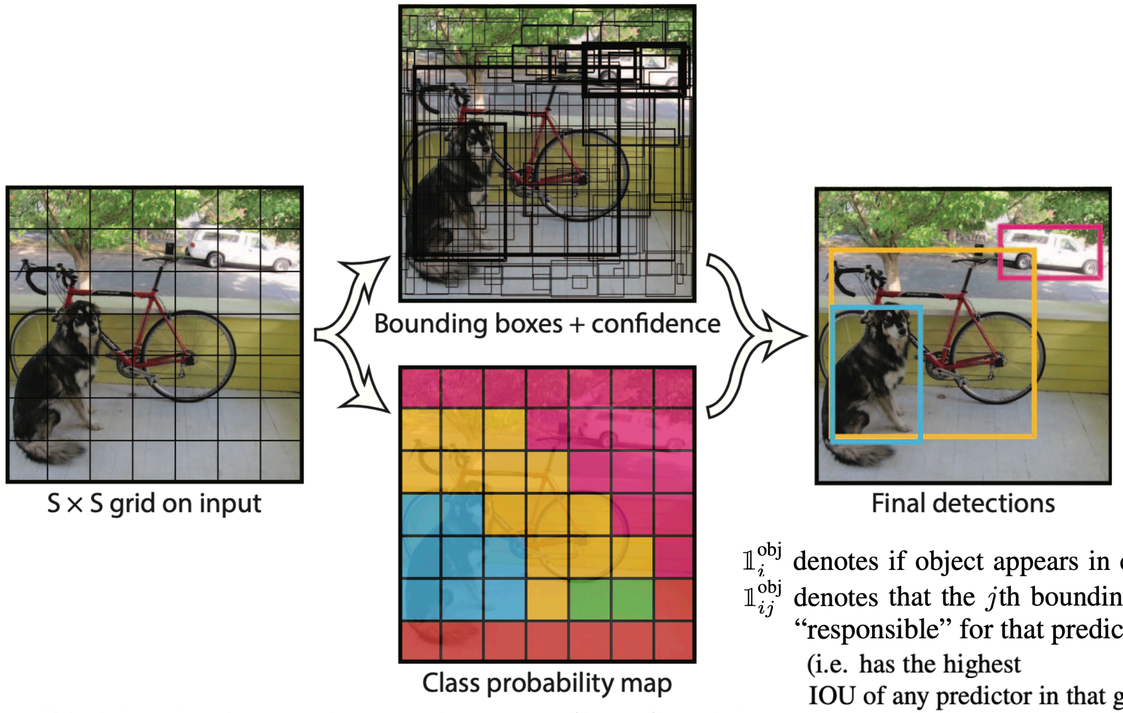
```
Compute match score using the sum of the distance
between centers of the two bounding boxes and the
intersection area of the boxes. box_merge computes
the average of the bounding boxes' coordinates
```


Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." *arXiv preprint arXiv:1312.6229* (2013).

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

YouTube Video





$\mathbb{1}_i^{\text{obj}}$ denotes if object appears in cell i
$\mathbb{1}_{ij}^{\text{obj}}$ denotes that the jth bounding box predictor in cell i is "responsible" for that prediction (i.e. has the highest IOU of any predictor in that grid cell)

$$\text{Confidence} \rightarrow \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

The confidence score reflects how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts.

$(x, y) \rightarrow$ center of the box relative to the bounds of the grid cell

$(w, h) \rightarrow$ width and height relative to the whole image

Each grid cell also predicts C conditional class probabilities, $\Pr(\text{Class}_i|\text{Object})$

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

class-specific confidence scores

$S \times S \times (B * 5 + C)$ tensor

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

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

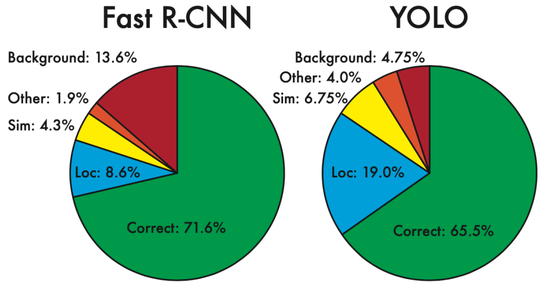
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

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

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

$\lambda_{\text{coord}} = 5$ and $\lambda_{\text{noobj}} = .5$

$S = 7$
$B = 2$
$C = 20$

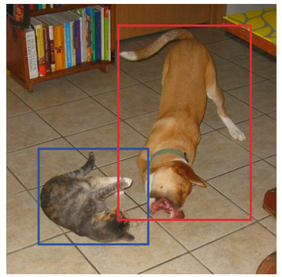


- Divide the input image into an S × S grid
- If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.
- Each grid cell predicts B bounding boxes and confidence scores for those boxes

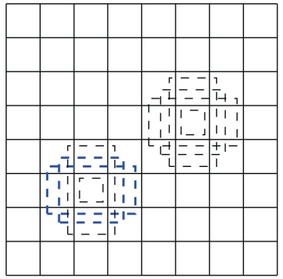
Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

SSD: Single Shot MultiBox Detector

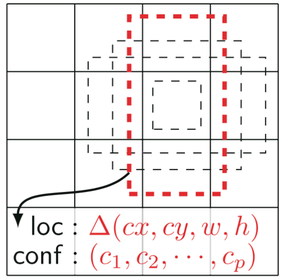
YouTube Video



(a) Image with GT boxes



(b) 8×8 feature map



(c) 4×4 feature map

$$8732 = 4 * 38 * 38 + 6 * 19 * 19 + 6 * 10 * 10 + 6 * 5 * 5 + 4 * 3 * 3 + 4 * 1 * 1$$

Training Objective

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

predicted box parameters; ground truth box parameters; number of matched default boxes

$L_{conf} \rightarrow$ softmax loss

$L_{loc} \rightarrow$ Smooth L1 loss

Scales and Aspect Ratios for Default Boxes

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

$s_{\max}$: scale of the highest layer; $s_{\min}$: scale of the lowest layer

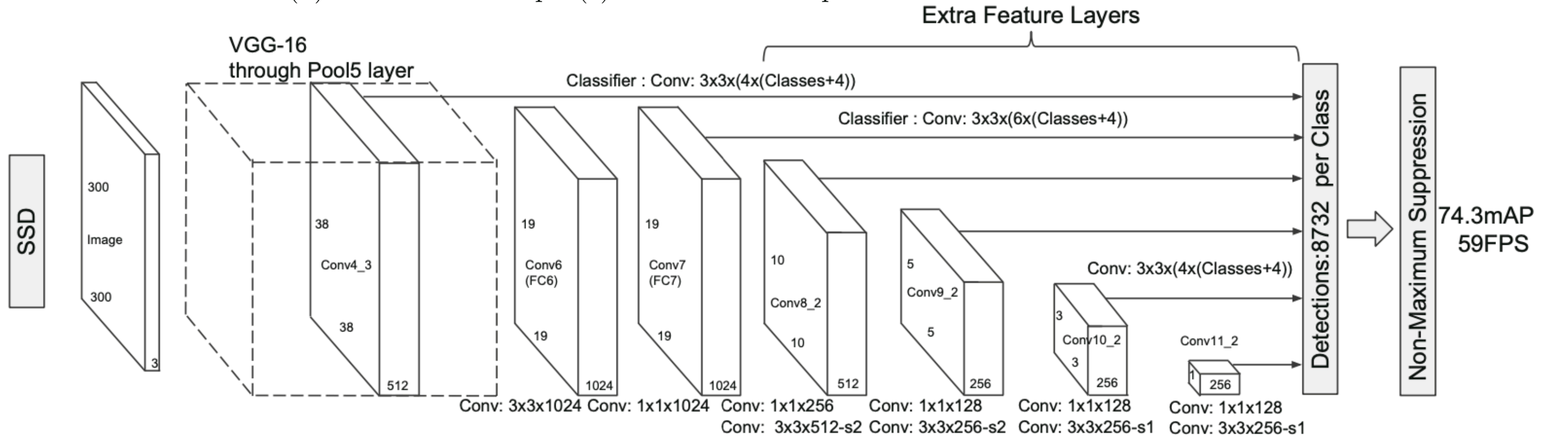
$a_r \in \{1, 2, 3, 1/2, 1/3\} \rightarrow$ aspect ratios

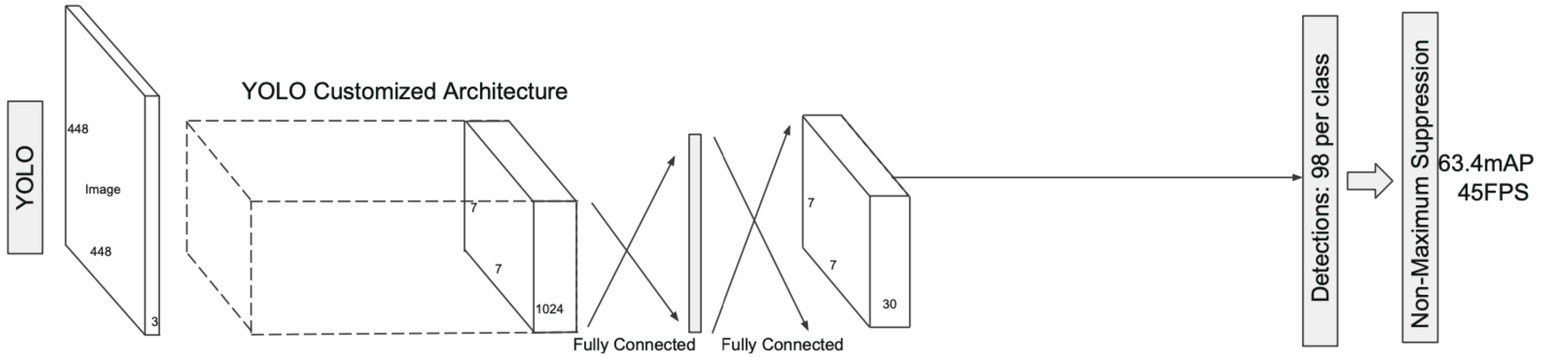
$w_k^a = s_k \sqrt{a_r} \rightarrow$ width

$h_k^a = s_k / \sqrt{a_r} \rightarrow$ height

$s'_k = \sqrt{s_k s_{k+1}} \rightarrow$ scale of the additional default box for aspect ratio 1

6 default boxes per feature map location





Method	mAP	FPS	Test batch size	# Boxes
Faster R-CNN [2] (VGG16)	73.2	7	1	300
Faster R-CNN [2] (ZF)	62.1	17	1	300
YOLO [5]	63.4	45	1	98
Fast YOLO [5]	52.7	155	1	98
SSD300	74.3	46	1	8732
SSD512	76.8	19	1	24564
SSD300	74.3	59	8	8732
SSD512	76.8	22	8	24564

Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, 2016.

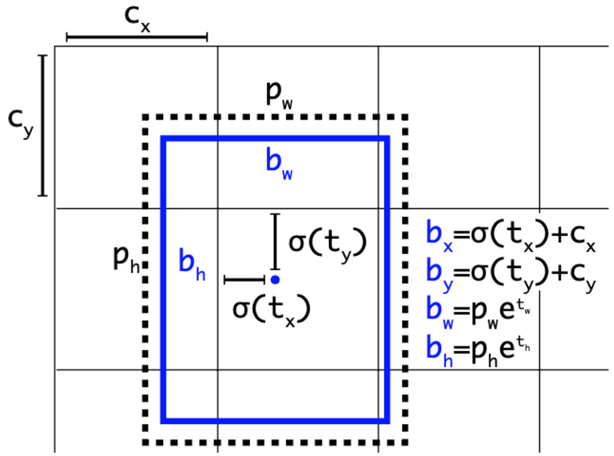
YOLO9000: Better, Faster, Stronger

YouTube Video

YOLO9000 can detect over 9000 object categories.

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

Direct location prediction

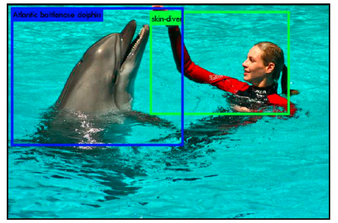


$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$
$$Pr(\text{object}) * IOU(b, \text{object}) = \sigma(t_o)$$

$(c_x, c_y) \rightarrow$ cell offset from top left corner of the image

$(t_x, t_y, t_w, t_h, t_o) \rightarrow$ predictions of the network

$(p_w, p_h) \rightarrow$ bounding box prior width & height

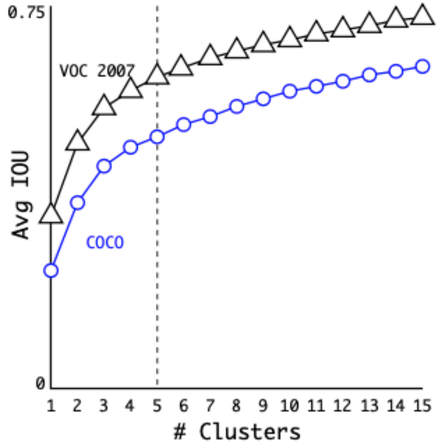


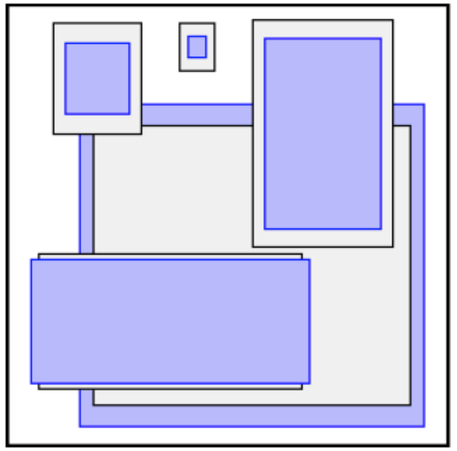
Joint classification and detection

See the paper!

Dimension Clusters

$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$$





k-means clustering on the training set box dimensions

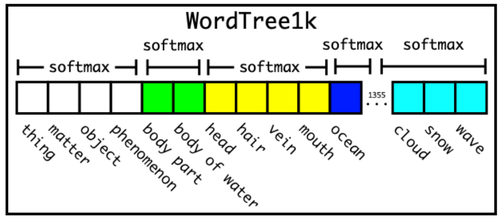
Passthrough Layer

Darknet-19

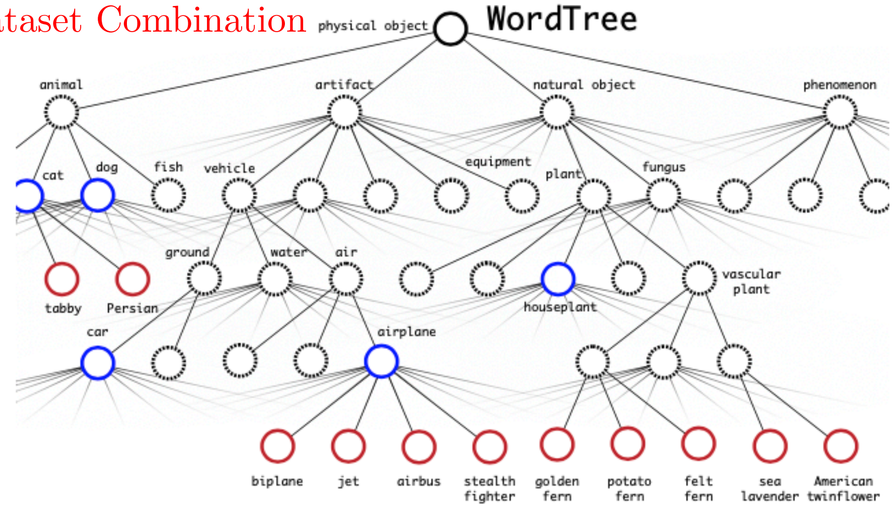
To detect small objects well, the 26×26×512 feature maps from earlier layer is mapped into 13×13×2048 feature map, then concatenated with the original 13×13 feature maps for detection.

Hierarchical classification

$$Pr(\text{Norfolk terrier}) = Pr(\text{Norfolk terrier}|\text{terrier}) * Pr(\text{terrier}|\text{hunting dog}) * \ldots * Pr(\text{mammal}|Pr(\text{animal}) * Pr(\text{animal}|\text{physical object})$$



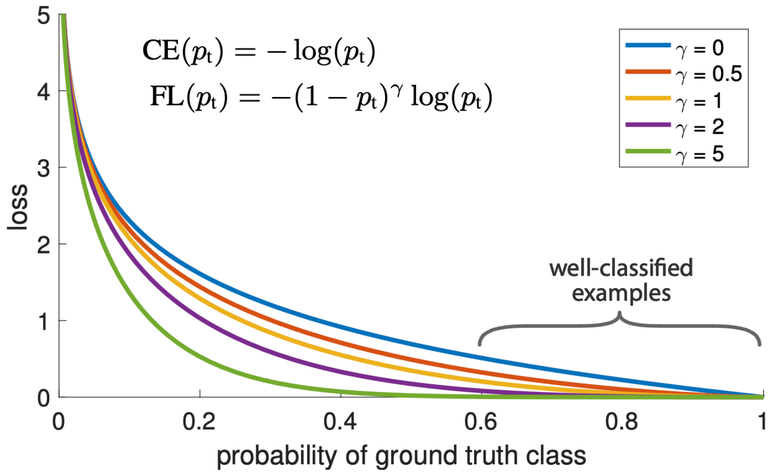
Dataset Combination

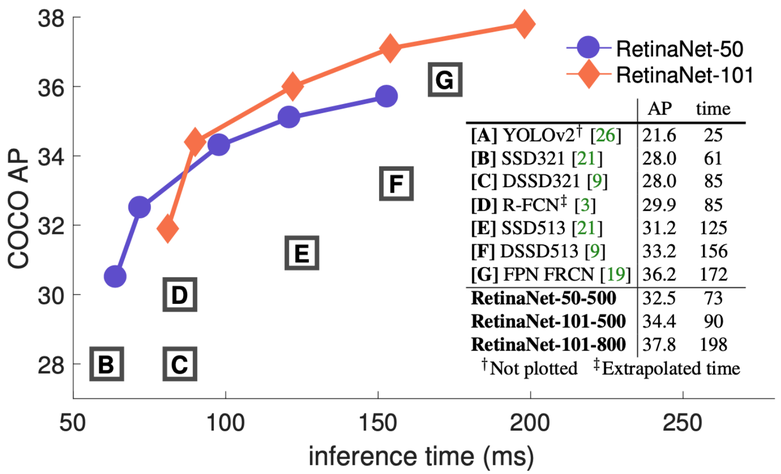


Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Focal Loss for Dense Object Detection

YouTube Video





	AP	time
[A] YOLOv2† [26]	21.6	25
[B] SSD321 [21]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [21]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [19]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

α	AP	AP_{50}	AP_{75}
.10	0.0	0.0	0.0
.25	10.8	16.0	11.7
.50	30.2	46.7	32.8
.75	31.1	49.4	33.0
.90	30.8	49.7	32.3
.99	28.7	47.4	29.9
.999	25.1	41.7	26.1

(a) **Varying α for CE loss** ($\gamma = 0$)

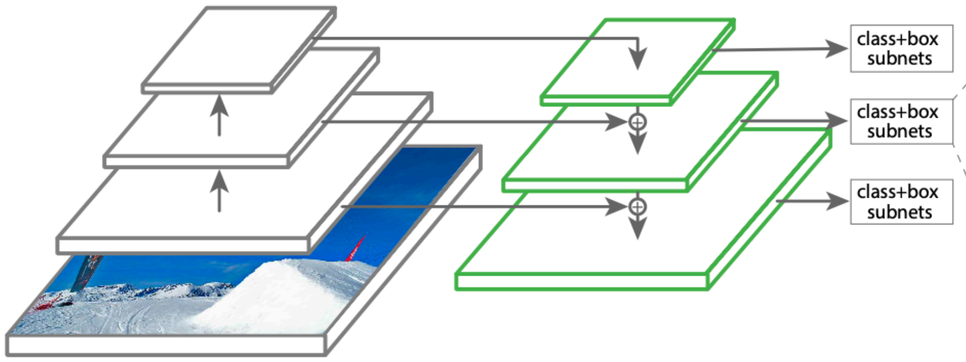
γ	α	AP	AP_{50}	AP_{75}
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

(b) **Varying γ for FL** (w. optimal α)

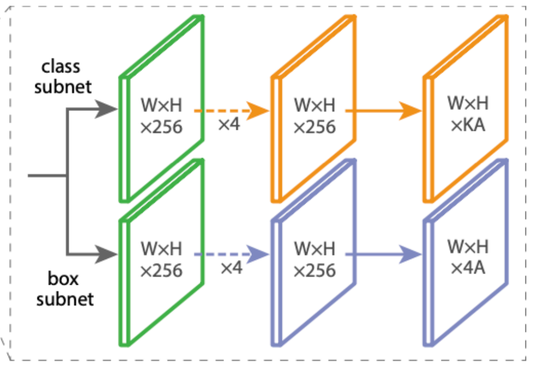
$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \rightarrow \alpha$-balanced focal loss

Identify class imbalance during training as the main obstacle impeding one-stage detector from achieving state-of-the-art accuracy and propose a new loss function that eliminates this barrier.

RetinaNet



(a) ResNet (b) feature pyramid net



(c) class subnet (top) (d) box subnet (bottom)

depth	scale	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	time
50	400	30.5	47.8	32.7	11.2	33.8	46.1	64
50	500	32.5	50.9	34.8	13.9	35.8	46.7	72
50	600	34.3	53.2	36.9	16.2	37.4	47.4	98
50	700	35.1	54.2	37.7	18.0	39.3	46.4	121
50	800	35.7	55.0	38.5	18.9	38.9	46.3	153
101	400	31.9	49.5	34.1	11.6	35.8	48.5	81
101	500	34.4	53.1	36.8	14.7	38.5	49.1	90
101	600	36.0	55.2	38.7	17.4	39.6	49.7	122
101	700	37.1	56.6	39.8	19.1	40.6	49.4	154
101	800	37.8	57.5	40.8	20.2	41.1	49.2	198

(e) **Accuracy/speed trade-off** RetinaNet (on `test-dev`)

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [15]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [19]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [16]	Inception-ResNet-v2 [33]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [31]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [26]	DarkNet-19 [26]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [21, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	**39.1**	**59.1**	**42.3**	**21.8**	**42.7**	50.2

Lin, Tsung-Yi, et al. "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.

Speed/Accuracy Trade-Offs For Modern Convolutional Object Detectors

The right speed/memory/accuracy balance for a given application and platform!

Self-driving cars: real-time performance

Mobile devices: small memory footprint

Faster R-CNN v.s. R-FCN v.s. SSD

Single model/single pass

Convolutional detection meta-architectures

$a \rightarrow$ anchor

$b \rightarrow$ best matching ground truth box (if one exists) corresponding to anchor a

$a \rightarrow$ positive anchor (if such a match exists)

$y_a \in \{1, 2, \ldots, K\} \rightarrow$ label class assigned to a

$\phi(b_a; a) \rightarrow$ vector encoding of box b w.r.t. anchor a

$\hookrightarrow$ box encoding

$$\phi(b_a; a) = [10\frac{x_c}{w_a}, 10\frac{y_c}{h_a}, 5\log w, 5\log h]$$

$x_c, y_c \rightarrow$ center coordinates

$w, h \rightarrow$ width and height

$w_a, h_a \rightarrow$ width and height of anchor a

$a \rightarrow$ negative anchor (if no match is found)

$y_a = 0$

$f_{\text{loc}}(\mathcal{I}; a, \theta) \rightarrow$ predicted box encoding for anchor a

$f_{\text{cls}}(\mathcal{I}; a, \theta) \rightarrow$ predicted class for anchor a

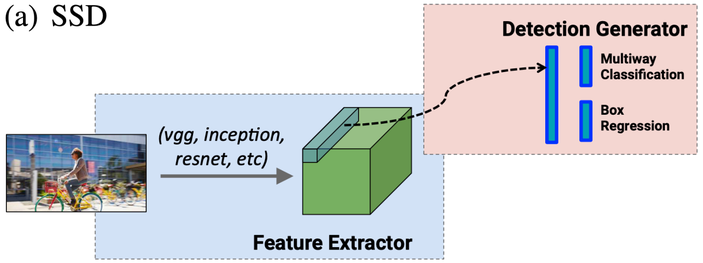
$\mathcal{I} \rightarrow$ image

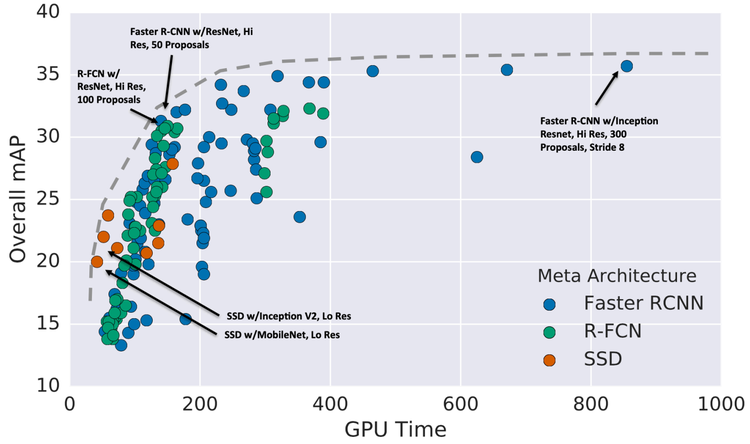
$\theta \rightarrow$ model parameters

$[x_0, y_0, x_1, y_1]$ are min/max coordinates of a box

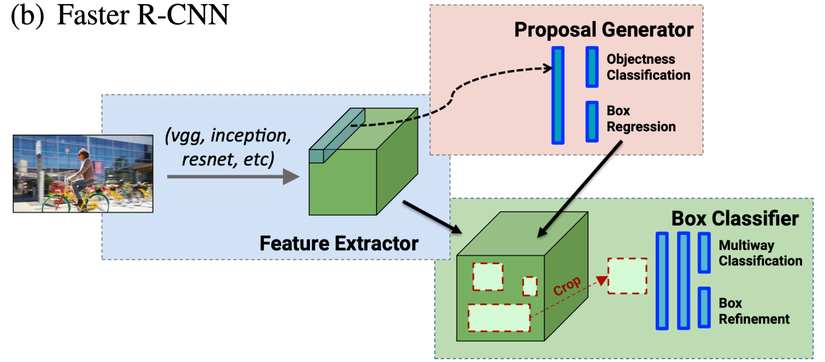
Paper	Meta-architecture	Feature Extractor	Matching	Box Encoding $\phi(b_a, a)$	Location Loss functions
Szegedy et al. [39]	SSD	InceptionV3	Bipartite	$[x_0, y_0, x_1, y_1]$	L_2
Redmon et al. [28]	SSD	Custom (GoogLeNet inspired)	Box Center	$[x_c, y_c, \sqrt{w}, \sqrt{h}]$	L_2
Ren et al. [30]	Faster R-CNN	VGG	Argmax	$[\frac{x_c}{w_a}, \frac{y_c}{h_a}, \log w, \log h]$	SmoothL_1
He et al. [13]	Faster R-CNN	ResNet-101	Argmax	$[\frac{x_c}{w_a}, \frac{y_c}{h_a}, \log w, \log h]$	SmoothL_1
Liu et al. [25] (v1)	SSD	InceptionV3	Argmax	$[x_0, y_0, x_1, y_1]$	L_2
Liu et al. [25] (v2, v3)	SSD	VGG	Argmax	$[\frac{x_c}{w_a}, \frac{y_c}{h_a}, \log w, \log h]$	SmoothL_1
Dai et al [6]	R-FCN	ResNet-101	Argmax	$[\frac{x_c}{w_a}, \frac{y_c}{h_a}, \log w, \log h]$	SmoothL_1

(a) SSD

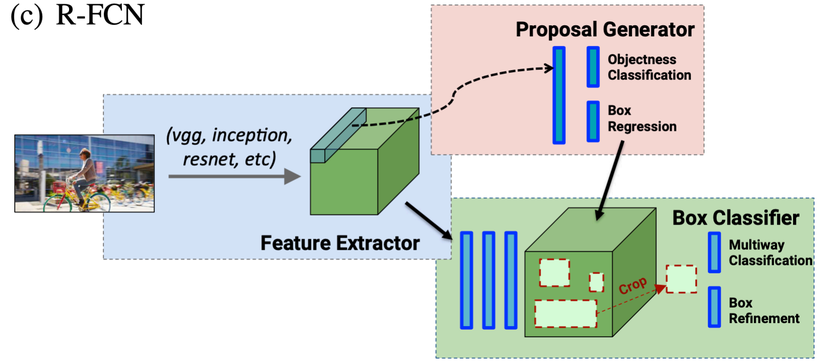




(b) Faster R-CNN



(c) R-FCN

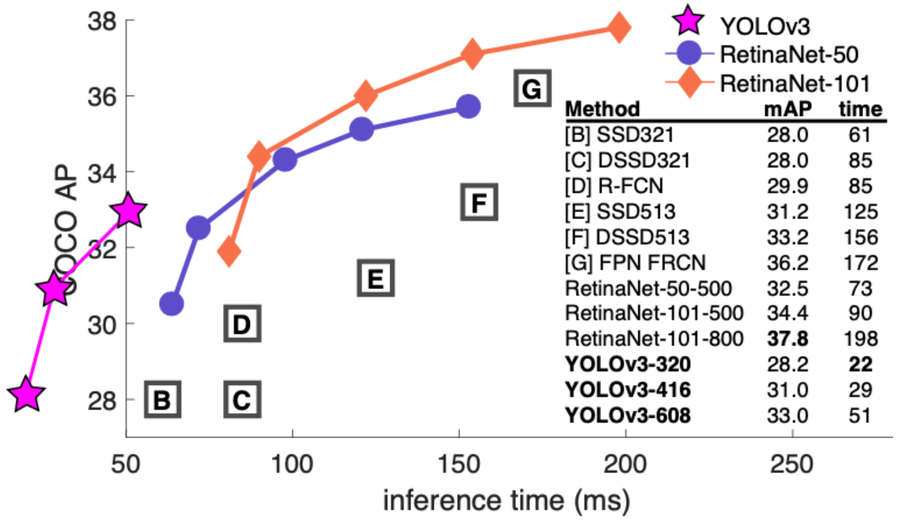


$$\mathcal{L}(a, \mathcal{I}; \theta) = \alpha \cdot \mathbb{1}[a \text{ is positive}] \cdot \ell_{loc}(\phi(b_a; a) - f_{loc}(\mathcal{I}; a, \theta)) + \beta \cdot \ell_{cls}(y_a, f_{cls}(\mathcal{I}; a, \theta))$$

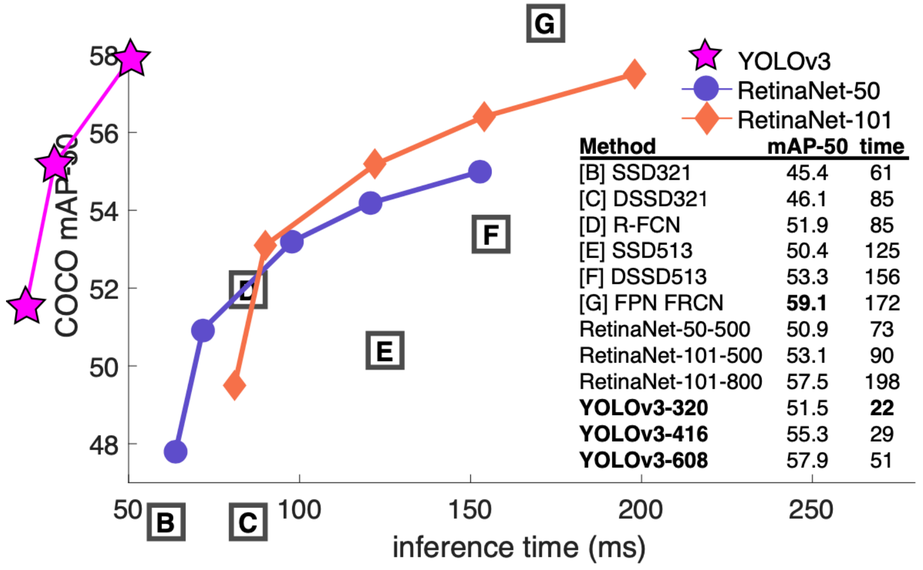
Huang, Jonathan, et al. "Speed/accuracy trade-offs for modern convolutional object detectors." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

YOLOv3: An Incremental Improvement

YouTube Video



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Darknet-53

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

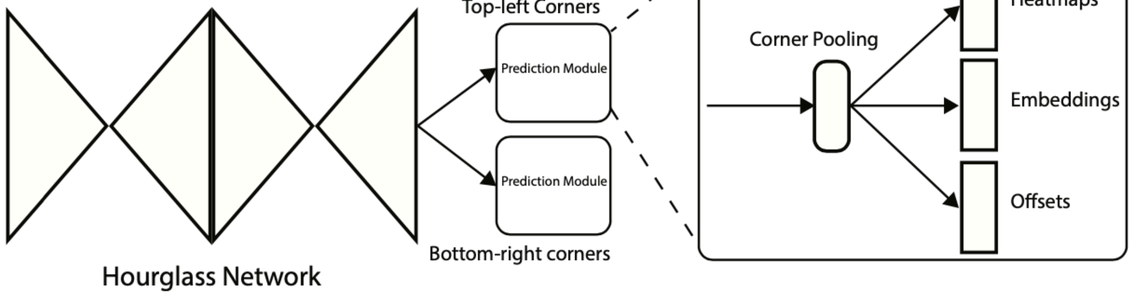
In experiments with COCO, the model predict 3 boxes at each scale (3 different scales similar to feature pyramid networks) so the tensor is NxNx[3*(4+1+80)] for the 4 bounding box offsets, 1 objectness prediction, and 80 class predictions.

Class prediction: Does not use a softmax, instead simply uses independent logistic classifiers. During training it uses binary cross-entropy loss for the class predictions.

Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

CornerNet: Detecting Objects as Paired Keypoints

Anchor box free!



Detecting Corners

Two sets of heatmaps:

– top-left corners (C channels of size $H \times W$)

– bottom-right corners (C channels of size $H \times W$)

$C \rightarrow$ number of categories (no background channel)

Each channel is a binary mask indicating the locations of the corners for a class.

$p_{cij} \rightarrow$ score at location (i, j) for class c in the predicted heatmaps

$y_{cij} \rightarrow$ "ground-truth" heatmap augmented with the unnormalized Gaussians

$$L_{det} = \frac{-1}{N} \sum_{c=1}^{C} \sum_{i=1}^{H} \sum_{j=1}^{W} \begin{cases} (1-p_{cij})^{\alpha} \log(p_{cij}) & \text{if } y_{cij} = 1 \\ (1-y_{cij})^{\beta} (p_{cij})^{\alpha} \log(1-p_{cij}) & \text{otherwise} \end{cases}$$

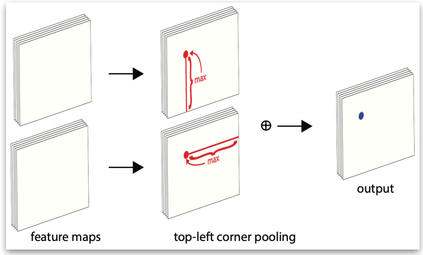
a variant of focal loss

$(1-y_{cij})^{\beta}$ reduces the penalty around the ground-truth locations

$$\boldsymbol{o}_k = \left(\frac{x_k}{n} - \left\lfloor \frac{x_k}{n} \right\rfloor, \frac{y_k}{n} - \left\lfloor \frac{y_k}{n} \right\rfloor \right) \rightarrow \text{offset for corner } k$$

$n \rightarrow$ downsampling factor

$$L_{off} = \frac{1}{N} \sum_{k=1}^{N} \text{SmoothL1Loss}(\boldsymbol{o}_k, \hat{\boldsymbol{o}}_k)$$



Grouping Corners

Associative Embedding (multi-person pose estimation)

$e_{t_k}, e_{b_k} \rightarrow$ embeddings of the top-left and bottom-right corners for object k

Group the corners:

$$L_{pull} = \frac{1}{N} \sum_{k=1}^{N} \left[(e_{t_k} - e_k)^2 + (e_{b_k} - e_k)^2 \right]$$

Separate the corners:

$$L_{push} = \frac{1}{N(N-1)} \sum_{k=1}^{N} \sum_{\substack{j=1 \\ j \neq k}}^{N} \max(0, \Delta - |e_k - e_j|)$$

$e_k = (e_{b_k} + e_{t_k})/2, \Delta = 1$

$$L = L_{det} + \alpha L_{pull} + \beta L_{push} + \gamma L_{off}$$

Corner Pooling

No local visual evidence for the presence of corners!

Law, Hei, and Jia Deng. "Cornernet: Detecting objects as paired keypoints." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

FCOS: Fully Convolutional One-Stage Object Detection

FCOS is anchor box free, as well as proposal free!

$F \in \mathbb{R}^{H \times W \times C}$ → feature maps at some layer of a backbone CNN

s → total stride until that layer

$\{B_i\}$ → ground truth bounding boxes for an inout image

$B_i = (\underbrace{x_0^i, y_0^i}_{\text{left-top}}, \underbrace{x_1^i, y_1^i}_{\text{right-bottom}}, \underbrace{c^i}_{\text{class}}) \in \mathbb{R}^4 \times \{1, 2, \ldots, C\}$

$C = 80$ for Ms-COCO

(x, y) → each location on the feature map

$(\lfloor \frac{s}{2} \rfloor + xs, \lfloor \frac{s}{2} \rfloor + ys)$ → (x, y) mapped back onto the input image

near the center of the receptive field of the location (x, y)

View locations as training samples, rather than the anchor boxes!

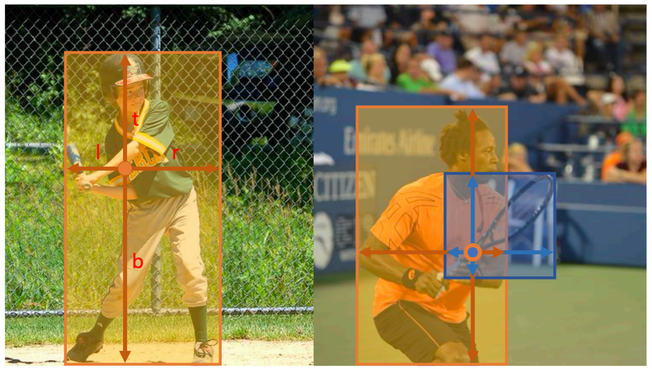
(x, y) is considered as a positive sample if it falls into any ground-truth box

c^* → ground truth label of location (x, y) (class label of the ground-truth box)

Otherwise, it is a negative sample and $c^* = 0$ (background)

$t^* = (l^*, t^*, r^*, b^*)$ → regression targets for the location

$l^* = x - x_0^i, t^* = y - y_0^i, r^* = x_1^i - x, b^* = y_1^i - y$



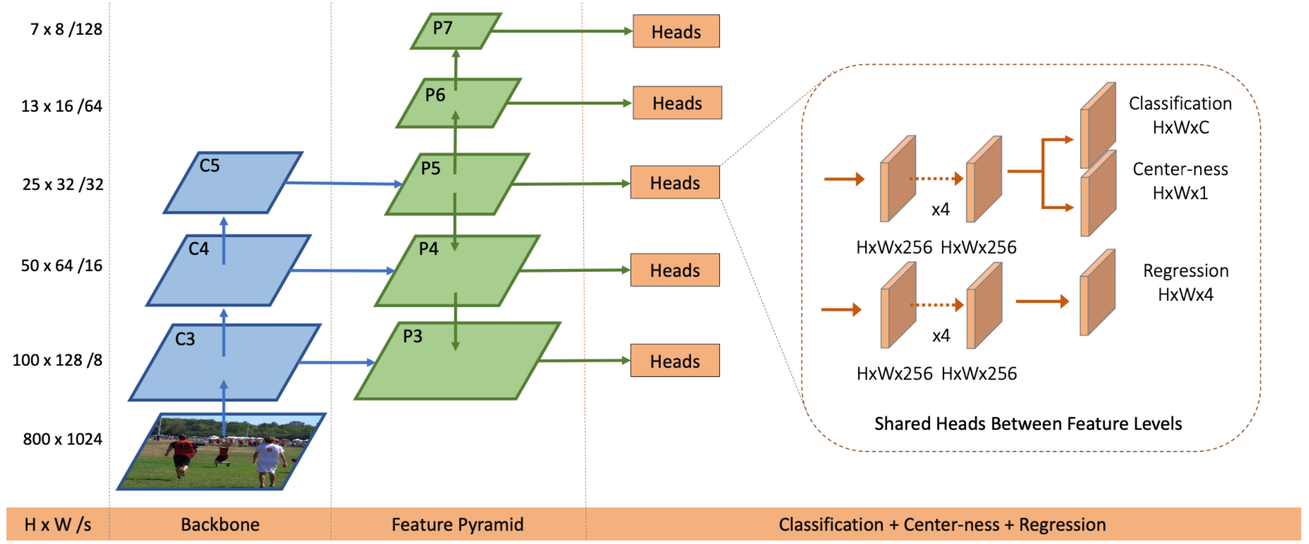
ambiguous sample: if a location falls into multiple bounding boxes

choose bounding box with minimal area as its regression target

multi-level prediction with FPN

Centerness

$$\text{Centerness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}$$



Network outputs

p → C dimensional vector of classification labels

t → 4 dimensional vector of bounding box coordinates

train C binary classifiers

Training loss

$$L(\{p_{x,y}\}, \{t_{x,y}\}) = \frac{1}{N_{\text{pos}}} \sum_{x,y} \underbrace{L_{\text{cls}}(p_{x,y}, c^*_{x,y})}_{\text{focal loss}} + \frac{\lambda}{N_{\text{pos}}} \sum_{x,y} \mathbb{1}_{\{c^*_{x,y} > 0\}} \underbrace{L_{\text{reg}}(t_{x,y}, t^*_{x,y})}_{\text{IoU loss}}$$

Inference

Choose locations (x, y) with $p_{x,y} > 0.05$ as positive samples!

$$-\log \frac{\text{Intersection}}{\text{Union}}$$

Multi-level prediction with FPN

A location is marked as not requiring to regress a bounding box, if:

$\max(l^*, t^*, r^*, b^*) > m_i$ or $\max(l^*, t^*, r^*, b^*) < m_{i-1}$

m_i → maximum distance that feature level i needs to regress

$\text{Centerness}^* \in (0, 1)$ → trained with binary cross entropy loss (BCE) → non-maximum suppression (NMS) using centerness times class score

Tian, Zhi, et al. "Fcos: Fully convolutional one-stage object detection." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

Objects as Points

CenterNet: Use keypoint estimation to find center points and regress to all other object properties, such as size, 3D location, orientation, and even pose.

Keypoint Estimation

$I \in \mathbb{R}^{W \times H \times 3} \rightarrow$ input image

$\hat{Y} \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C} \rightarrow$ keypoint heatmap prediction

$R \rightarrow$ output stride (i.e., downsampled by a factor of R)

$C \rightarrow$ number of keypoint types

Human joints in pose estimation ($C = 17$) and object categories in object detection ($C = 80$)

$\hat{Y}_{x,y,c} = 1 \rightarrow$ corresponds to a detected keypoint

$\hat{Y}_{x,y,c} = 0 \rightarrow$ corresponds to the background

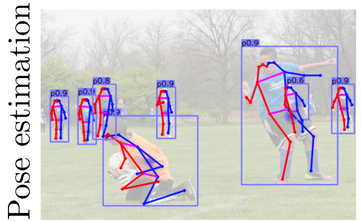
$p \in \mathbb{R}^2 \rightarrow$ ground truth keypoint of class c

$\tilde{p} = \lfloor \frac{p}{R} \rfloor \rightarrow$ low-resolution equivalent of p

$Y_{xyc} = \exp\left(-\frac{(x-\tilde{p}_x)^2+(y-\tilde{p}_y)^2}{2\sigma_p^2}\right) \rightarrow$ Gaussian kernel

$\sigma_p \rightarrow$ object-size adaptive standard deviation

$Y \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C} \rightarrow$ ground truth heatmap



If two Gaussians of the same class overlap, take the element-wise maximum.

Penalty-reduced pixel-wise logistic regression with focal loss:

$$L_k = \frac{-1}{N} \sum_{xyc} \begin{cases} (1-\hat{Y}_{xyc})^{\alpha} \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\ (1-Y_{xyc})^{\beta} (\hat{Y}_{xyc})^{\alpha} & \text{otherwise} \\ \log(1-\hat{Y}_{xyc}) & \end{cases}$$

$N \rightarrow$ number of keypoints in image

$\hat{O} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2} \rightarrow$ local offset for each center-point

$$L_{off} = \frac{1}{N} \sum_{p} \left| \hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p} \right) \right|$$

recover the discretization error caused by the output stride

Objects as Points

$(x_1^{(k)}, y_1^{(k)}, x_2^{(k)}, y_2^{(k)}) \rightarrow$ bounding box of object k with category c_k

$p_k = \frac{1}{2}(x_1^{(k)} + x_2^{(k)}, y_1^{(k)} + y_2^{(k)}) \rightarrow$ center point of object k

Use the keypoint estimator $\hat{Y}$ to predict all center-points

$s_k = (x_2^{(k)} - x_1^{(k)}, y_2^{(k)} - y_1^{(k)}) \rightarrow$ size of object k

$\hat{S} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2} \rightarrow$ single size prediction of all object categories

$$L_{size} = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{S}_{p_k} - s_k \right|$$

$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off}$$

$[\hat{Y}, \hat{O}, \hat{S}] \rightarrow C + 4$ network outputs at each location

keypoint heatmap [C]

local offset [2]

object size [2]

Zhou, Xingyi, Dequan Wang, and Philipp Krähenbühl. "Objects as points." *arXiv preprint arXiv:1904.07850* (2019).

EfficientDet: Scalable and Efficient Object Detection

Boulder

Multi-Scale Feature Fusion

aggregate features at different resolutions

$P^{\text{in}} = (P^{\text{in}}_{l_1}, P^{\text{in}}_{l_2}, \ldots) \rightarrow$ list of multi-scale features

$P^{\text{in}}_{l_i} \rightarrow$ feature at level l_i

$P^{\text{out}} = f(P^{\text{in}}) \rightarrow$ design f

FPN (Feature Pyramid Network)

$P^{\text{in}} = (P^{\text{in}}_3, P^{\text{in}}_4, \ldots, P^{\text{in}}_7)$

$P^{\text{in}}_i \rightarrow$ feature level with resolution $\frac{1}{2^i}$ of the input image

$$P^{\text{out}}_7 = \text{Conv}(P^{\text{in}}_7)$$

$$P^{\text{out}}_6 = \text{Conv}(P^{\text{in}}_6 + \text{Resize}(P^{\text{out}}_7))$$

$$\vdots$$

$$P^{\text{out}}_3 = \text{Conv}(P^{\text{in}}_3 + \text{Resize}(P^{\text{out}}_4))$$

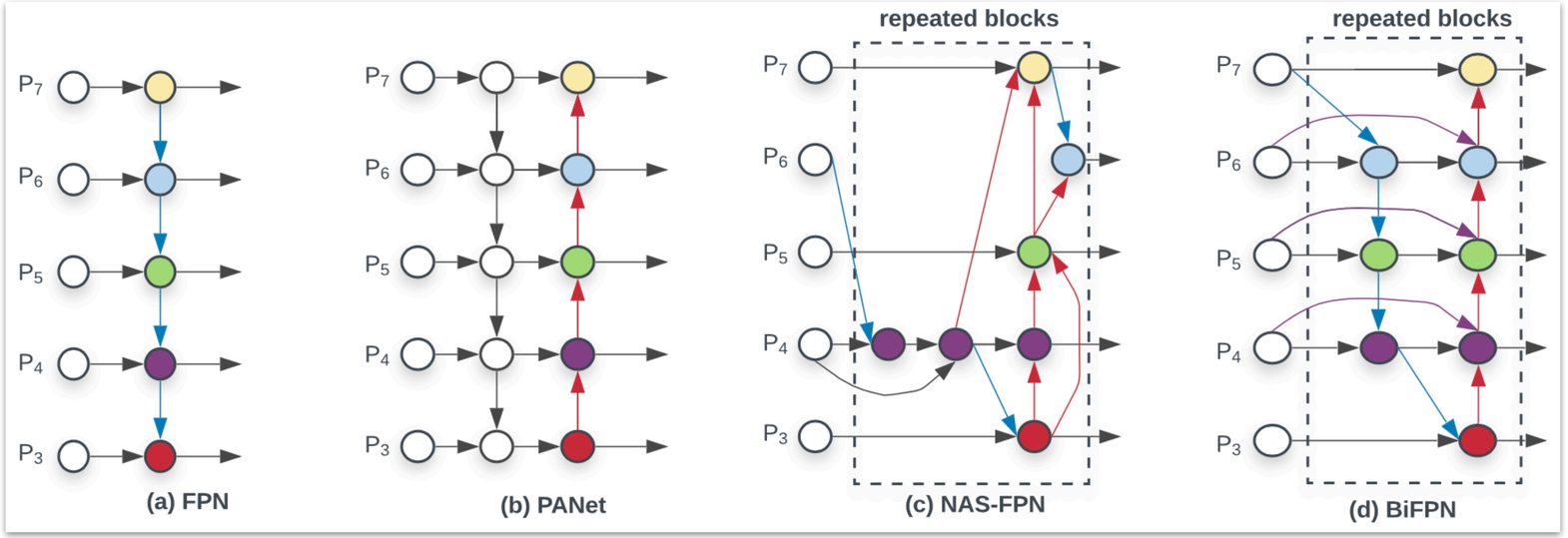
Weighted Bi-directional FPN (BiFPN)

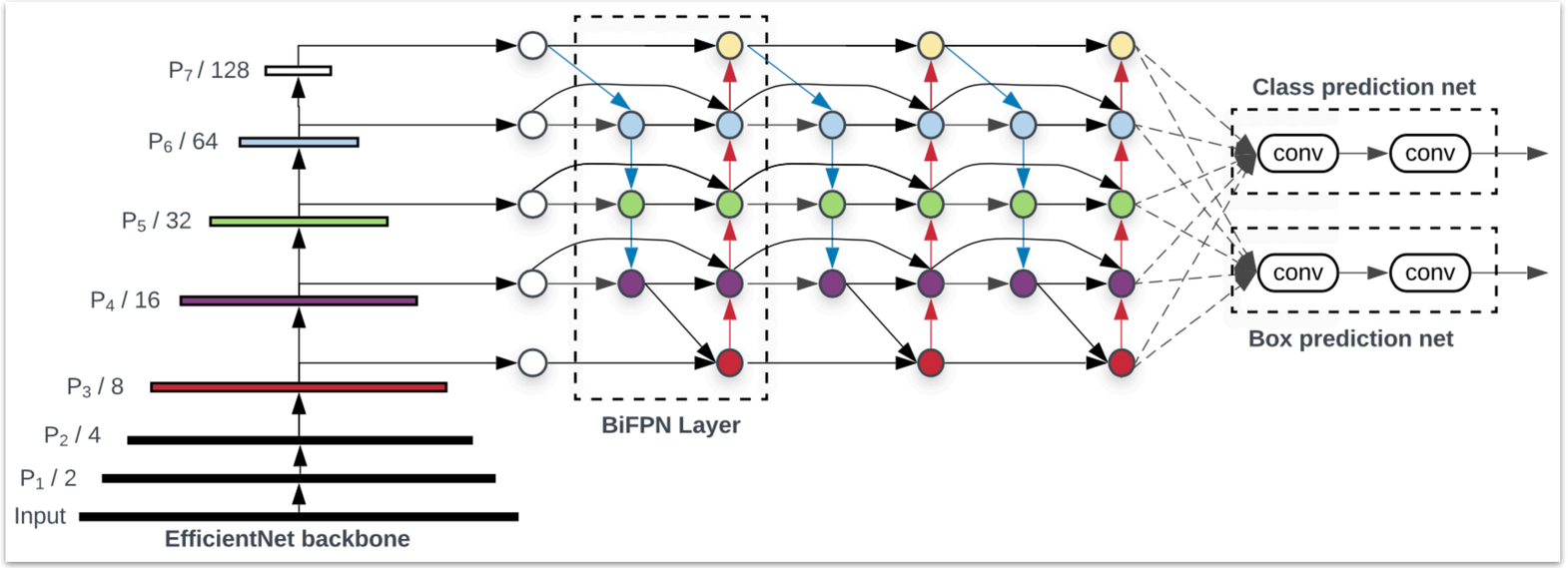
$$P^{\text{td}}_6 = \text{Conv}\left(\frac{w_1 P^{\text{in}}_6 + w_2 \text{Resize}(P^{\text{out}}_7)}{w_1 + w_2 + \varepsilon}\right)$$

$$P^{\text{out}}_6 = \text{Conv}\left(\frac{w'_1 P^{\text{in}}_6 + w'_2 P^{\text{td}}_6 + w'_3 \text{Resize}(P^{\text{out}}_5)}{w'_1 + w'_2 + w'_3 + \varepsilon}\right)$$

	Input size R_{input}	Backbone Network	BiFPN #channels W_{bifpn}	BiFPN #layers D_{bifpn}	Box/class #layers D_{class}
D0 ($\phi = 0$)	512	B0	64	3	3
D1 ($\phi = 1$)	640	B1	88	4	3
D2 ($\phi = 2$)	768	B2	112	5	3
D3 ($\phi = 3$)	896	B3	160	6	4
D4 ($\phi = 4$)	1024	B4	224	7	4
D5 ($\phi = 5$)	1280	B5	288	7	4
D6 ($\phi = 6$)	1280	B6	384	8	5
D6 ($\phi = 7$)	1536	B6	384	8	5

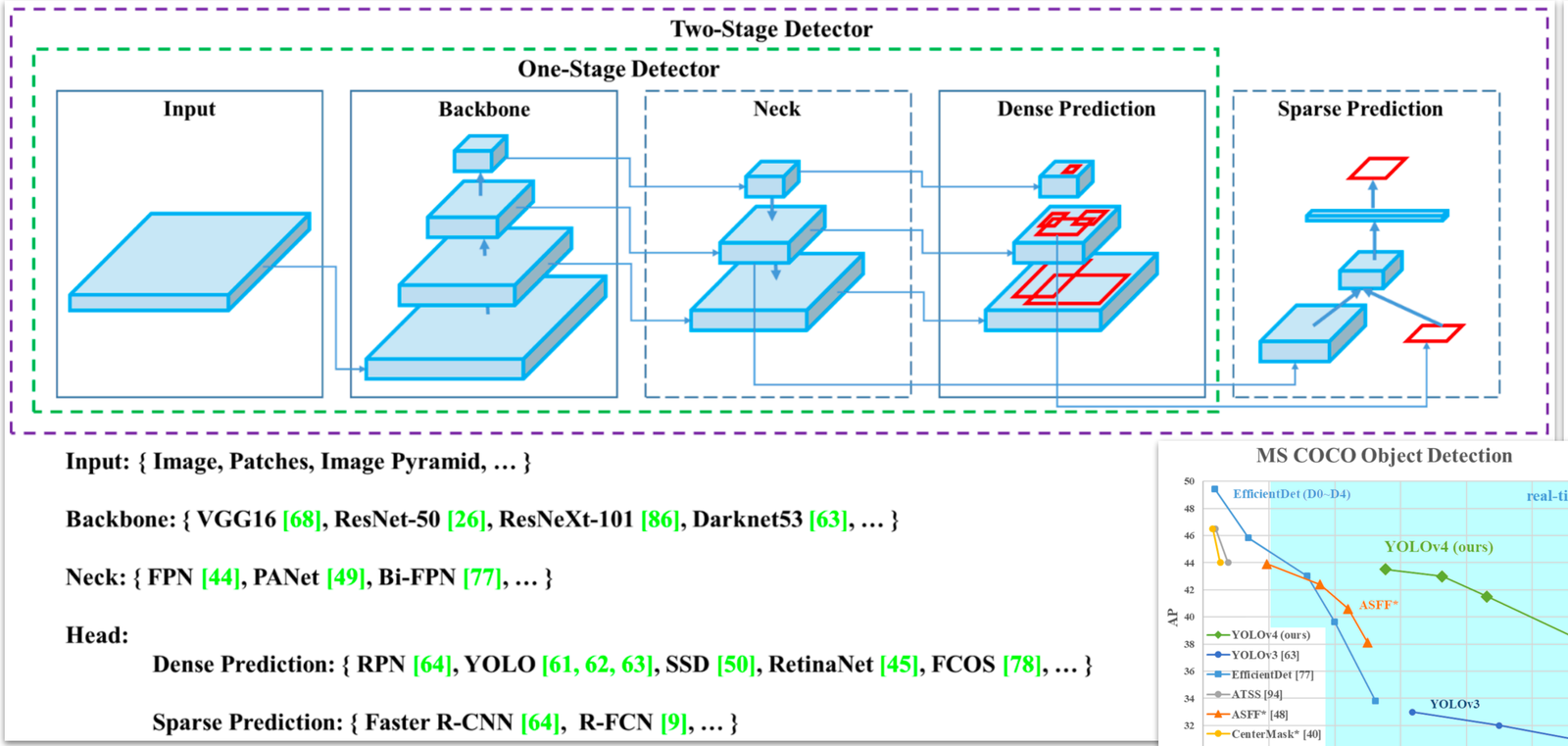
$\phi \rightarrow$ Compound Scaling Coefficient

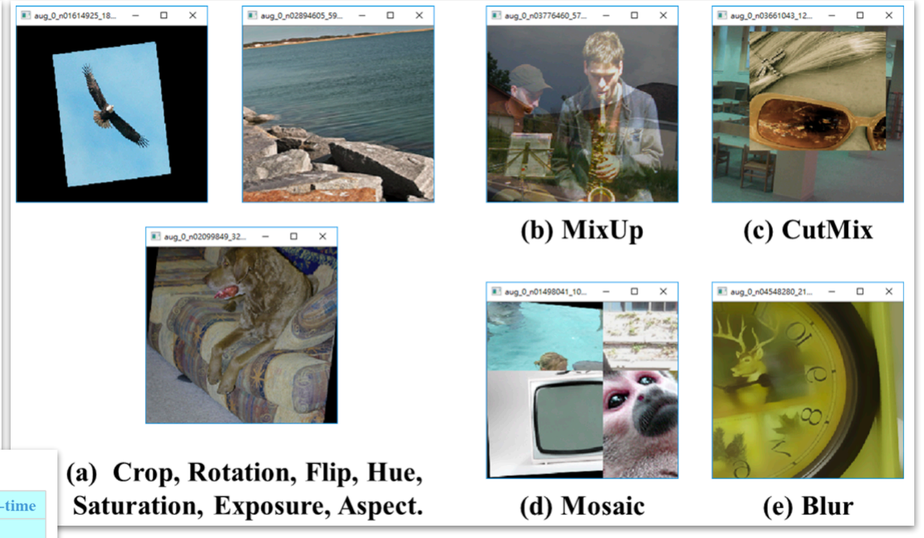




Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "Efficientdet: Scalable and efficient object detection." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

YOLOv4: Optimal Speed and Accuracy of Object Detection





Influence of BoF and Mish on the CSPDarknet-53 classifier accuracy.

MixUp	CutMix	Mosaic	Bluring	Label Smoothing	Swish	Mish	Top-1	Top-5
							77.2%	93.6%
	✓	✓		✓			**77.8%**	**94.4%**
	✓	✓		✓		✓	**78.7%**	**94.8%**

YOLOv4 consists of:

– Backbone: CSPDarknet53 (Cross-Stage-Partial-connections)

– Neck: SPP (Spatial Pyramid Pooling), PAN (Path Aggregation Network)

– Head: YOLOv3

YOLO v4 uses:

– Bag of Freebies (BoF) for backbone: CutMix and Mosaic data augmentation, DropBlock regularization, Class label smoothing

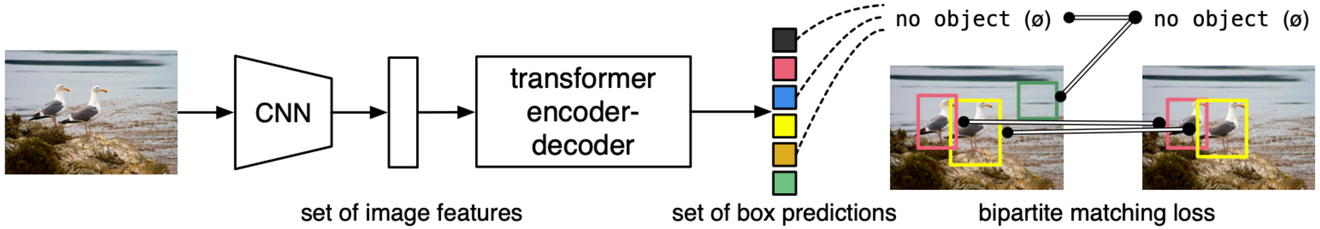
– Bag of Specials (BoS) for backbone: Mish activation, Cross-stage partial connections (CSP), Multi-input weighted residual connections (MiWRC)

– Bag of Freebies (BoF) for detector: CIoU-loss, CmBN (Cross mini-Batch Normalization), DropBlock regularization, Mosaic data augmentation, Self-Adversarial Training, Eliminate grid sensitivity, Using multiple anchors for a single ground truth, Cosine annealing scheduler, Optimal hyper-parameters, Random training shapes

– Bag of Specials (BoS) for detector: Mish activation, SPP-block, SAM-block (Spatial Attention Module), PAN path-aggregation block, DIoU-NMS (Non-Maximum Suppression)

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).

End-to-End Object Detection with Transformers



$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} [-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)})]$$

Down-weight the log-probability term when $c_i = \emptyset$ by a factor of 10 to account for class imbalances

Object detection set prediction loss

$N \rightarrow$ number of predictions

$y \rightarrow$ ground truth set of objects

$\hat{y} = \{\hat{y}_i\}_{i=1}^{N} \rightarrow$ set of N predictions

$N >$ # of objects in the image

$y \rightarrow$ set of size N padded by $\emptyset$ (no object)

$\sigma \in \mathcal{S}_N \rightarrow$ permutation of N elements

$$\hat{\sigma} = \arg\min_{\sigma \in \mathcal{S}_N} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) \rightarrow \text{Hungarian Algorithm}$$

$\mathcal{L}_{\text{match}}$ ← pairwise matching cost

$y_i = (c_i, b_i)$

$c_i \rightarrow$ target class label (which may be $\emptyset$)

$b_i \in [0, 1]^4 \rightarrow$ ground-truth bonding box center coordinates, height and width relative to the image size

$\hat{p}_{\sigma(i)}(c_i) \rightarrow$ probability of class c_i for the prediction with index $\sigma(i)$

$\hat{b}_{\sigma(i)} \rightarrow$ predicted box for the prediction with index $\sigma(i)$

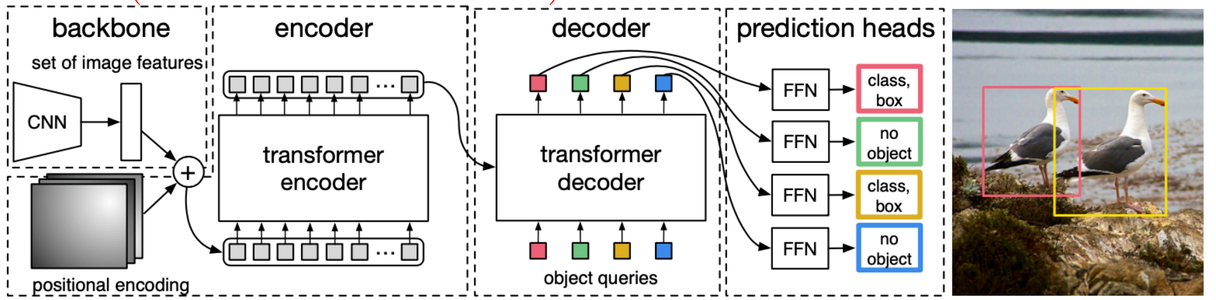
$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\mathbb{1}_{\{c_i \neq \emptyset\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$$

$$\mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) = \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\text{L1}} \|b_i - \hat{b}_{\sigma(i)}\|_1$$

$$\mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) = 1 - \left(\frac{|b_i \cap \hat{b}_{\sigma(i)}|}{|b_i \cup \hat{b}_{\sigma(i)}|} - \frac{|B(b_i, \hat{b}_{\sigma(i)}) \setminus b_i \cup \hat{b}_{\sigma(i)}|}{|B(b_i, \hat{b}_{\sigma(i)})|} \right)$$

$B(b_i, \hat{b}_{\sigma(i)})$ ← largest box containing b_i & $\hat{b}_{\sigma(i)}$

DETR (DEtection TRansformer) architecture



Object Querries → Positional Ecoding

$x_{\text{image}} \in \mathbb{R}^{3 \times H \times W}$

$f \in \mathbb{R}^{C \times \frac{H}{32} \times \frac{W}{32}}$

f ← low-resolution feature map

$C = 2048$

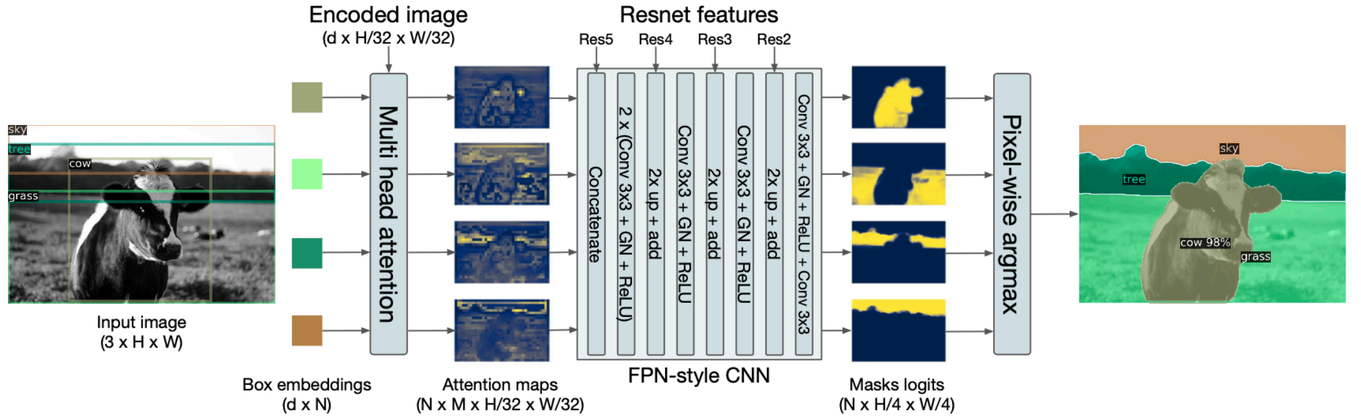
$f \triangleright 1 \times 1$ conv $\rightarrow z_0 \in \mathbb{R}^{d \times H \times W}$

$d < C$

$z_0 \in \mathbb{R}^{d \times (H/32) \cdot (W/32)} \rightarrow$ reshape

reshape ← input to the transformer encoder

DETR for panoptic segmentation



Carion, Nicolas, et al. "End-to-End Object Detection with Transformers." *arXiv preprint arXiv:2005.12872* (2020).

Deformable DETR: Deformable Transformers for End-to-End Object Detection

DETR eliminates the need for many hand-crafted components, e.g., anchor generation, rule-based training target assignment, non-maximum suppression (NMS) post-processing.

DETR has its own issues: (1) Small object detection requires higher resolution (multi-resolution) feature maps. However, the quadratic cost of transformer encoder is prohibitive. (2) DETR requires many more training epochs to converge compared to modern object detectors (because of uniformly spread attention initially during training).

Method	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	params	FLOPs	Training GPU hours	Inference FPS
Faster R-CNN + FPN	109	42.0	62.1	45.5	26.6	45.4	53.4	42M	180G	380	26
DETR	500	42.0	62.4	44.2	20.5	45.8	61.1	41M	86G	2000	28
DETR-DC5	500	43.3	63.1	45.9	22.5	47.3	61.1	41M	187G	7000	12
DETR-DC5	50	35.3	55.7	36.8	15.2	37.5	53.6	41M	187G	700	12
DETR-DC5$^+$	50	36.2	57.0	37.4	16.3	39.2	53.9	41M	187G	700	12
Deformable DETR	50	43.8	62.6	47.7	26.4	47.1	58.0	40M	173G	325	19
+ iterative bounding box refinement	50	45.4	64.7	49.0	26.8	48.3	61.7	40M	173G	325	19
++ two-stage Deformable DETR	50	46.2	65.2	50.0	28.8	49.2	61.7	40M	173G	340	19

Multi-scale Deformable Attention Module

$x^l \in \mathbb{R}^{C \times H_l \times W_l}, l = 1, \ldots, L \rightarrow$ input feature maps extracted by a CNN backbone at multiple scales

$z_q \in \mathbb{R}^C \rightarrow$ feature vector of query element q

$\hat{p}_q \in [0,1]^2 \rightarrow$ normalized coordinates of the reference point for each query element q

$$\text{MSDeformAttn}(z_q, \hat{p}_q, \{x^l\}_{l=1}^L) = \sum_{m=1}^{M} W_m \Big[\sum_{l=1}^{L} \sum_{k=1}^{K} \underbrace{A_{mlqk}}_{\text{attention weight}} \overbrace{W'_m x^l(\underbrace{\phi_l(\hat{p}_q)}_{p_{lq}} + \underbrace{\Delta p_{mlqk}}_{\text{offset}})}^{\text{interpolation}} \Big]$$

attention heads | multi-scale | sampled keys

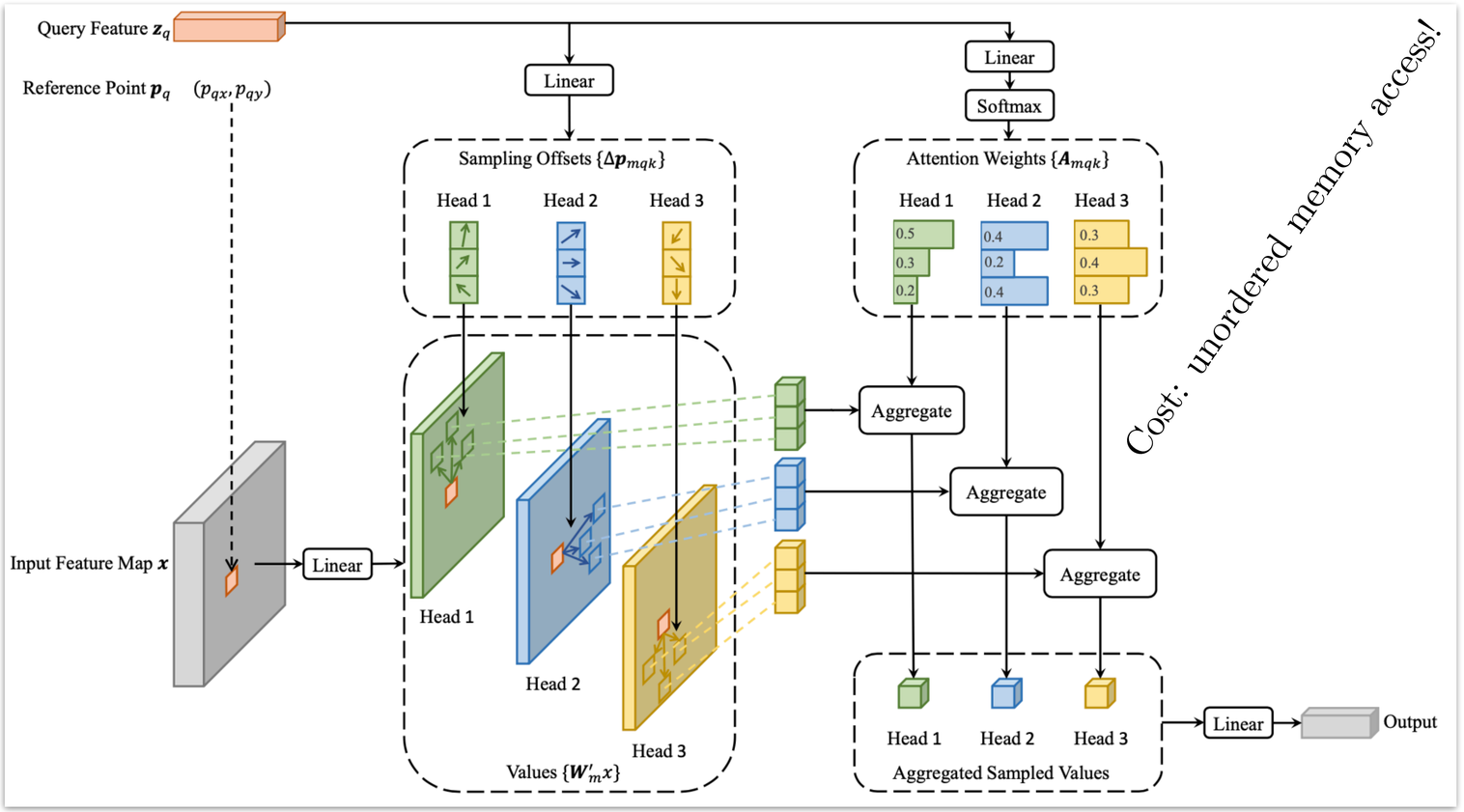
$K \ll HW \rightarrow$ total number of sampled keys

$N_q \rightarrow$ number of query points

$O(2N_qC^2 + \min(HWC^2, N_qKC^2)) \rightarrow$ complexity of the deformable attention module (i.e., $L = 1$)

$O(HWC^2) \rightarrow$ DETR encoder ($N_q = HW$)

$O(NKC^2) \rightarrow$ DETR decoder ($N_q = N$)



Zhu, Xizhou, et al. "Deformable detr: Deformable transformers for end-to-end object detection." *arXiv preprint arXiv:2010.04159* (2020).

Boulder

Questions?

YouTube Playlist