

FCOS: Fully Convolutional One-Stage Object Detection

EVOCARGO



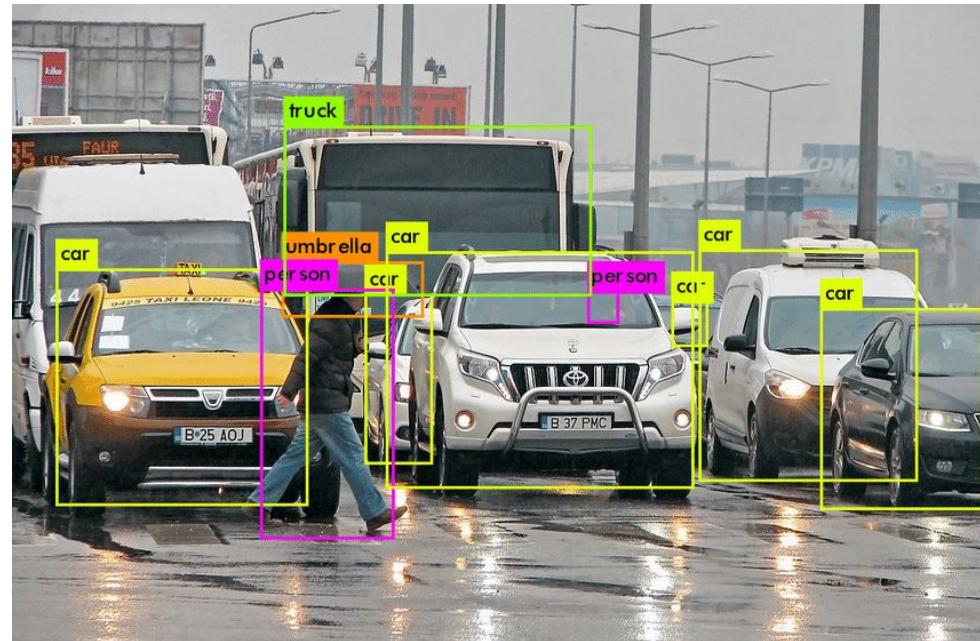
**girafe
ai**



Object detection

Задачи:

- Построить bounding box
- Классифицировать объект

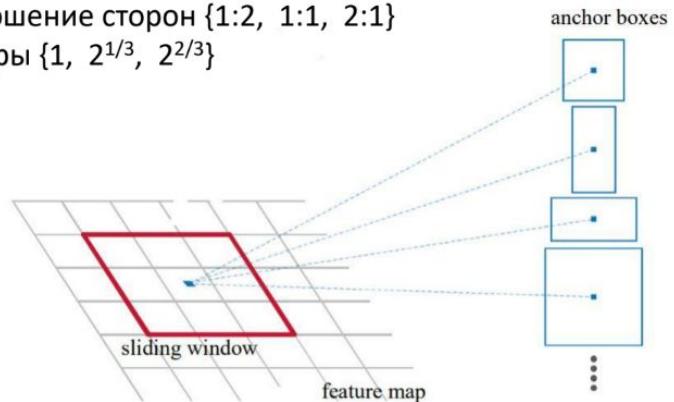


Проблемы с anchor box

- Чувствительность к гиперпараметрам
- Ограничение обобщающей способности модели
- Дисбаланс положительных и отрицательных anchor box'ов
- Необходим подсчет IoU для каждого anchor box'a

Anchor box:

- соотношение сторон {1:2, 1:1, 2:1}
- Размеры {1, $2^{1/3}$, $2^{2/3}$ }



Решение

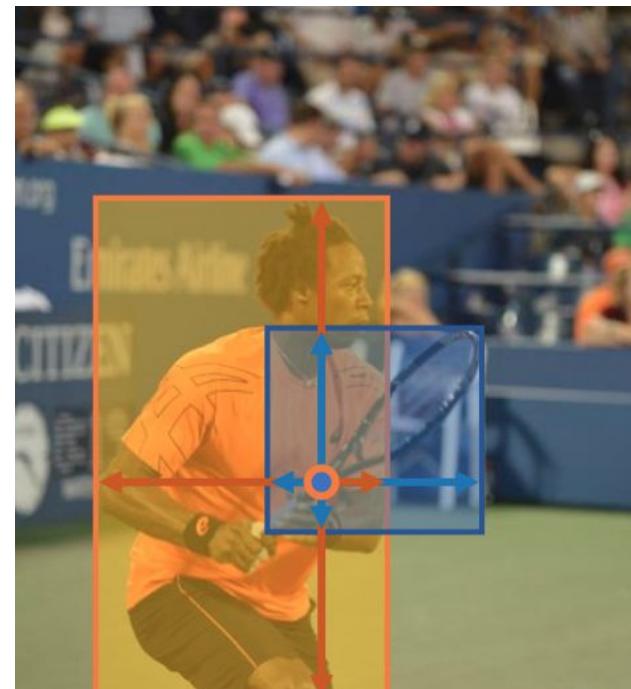
Per-pixel prediction подход

- Унифицированная среда
- Сокращения числа гиперпараметров
- Достижение SOTA результатов
- Переиспользование в других типах задач



Предшествующие подходы

- **DenseBox** - Baidu Research [2015]. Плохое качество
- **YOLO** - University of Washington, Allen Institute for AI, Facebook AI Research [2015]. Низкий recall
- **CornerNet** - Princeton University [2019]. Сложный постпроцессинг

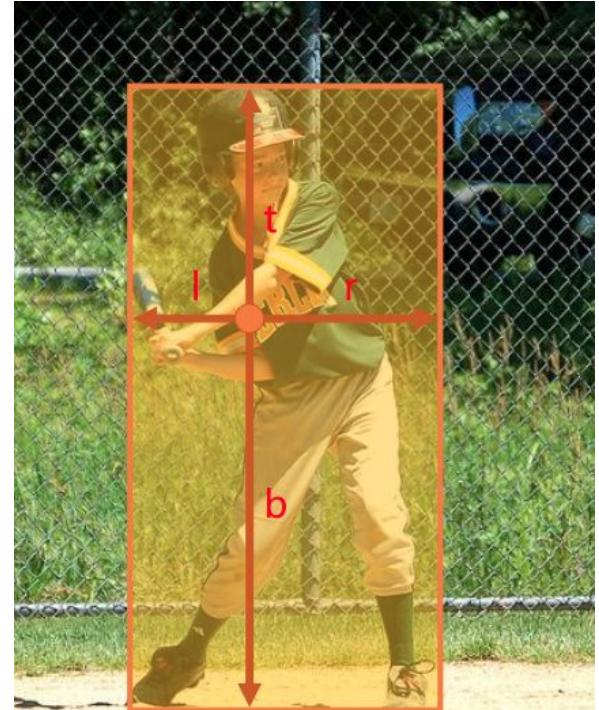


Модель

Построение карт признаков по принципу **FPN**

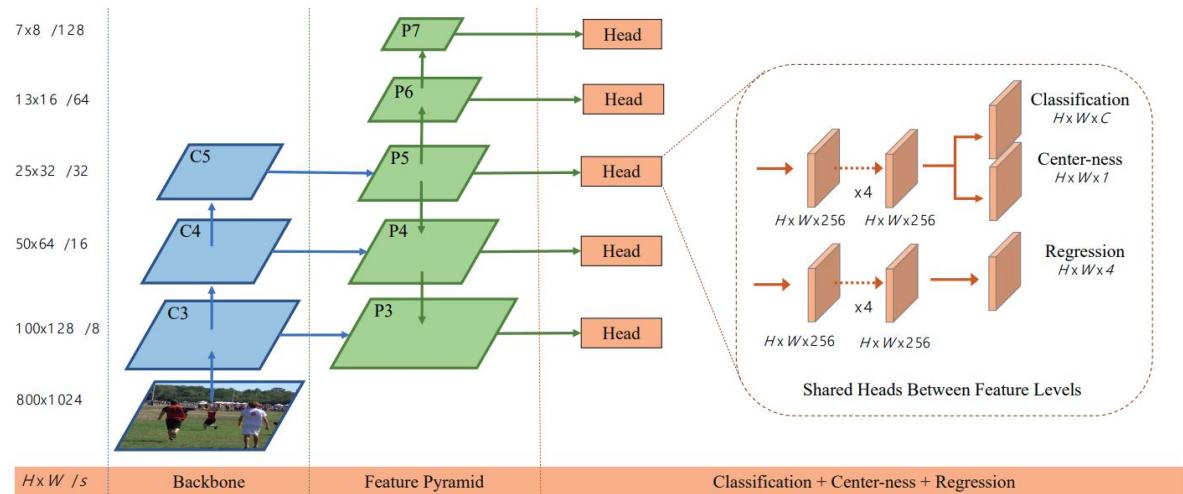
Для каждой точки (x, y) с карты признаков F_i ставится в соответствие точка на оригинальном изображении:
 $(\lfloor \frac{s}{2} \rfloor + xs, \lfloor \frac{s}{2} \rfloor + ys)$, s - число шагов до F_i

(l, t, r, b) - расстояние до границ bbox



Архитектура

- FPN подход для решения проблемы двузначности
- Для каждого уровня задается для граница принадлежности точки к объекту
- Обмен Head уровнями
- ResNeXt-64 в качестве backbone



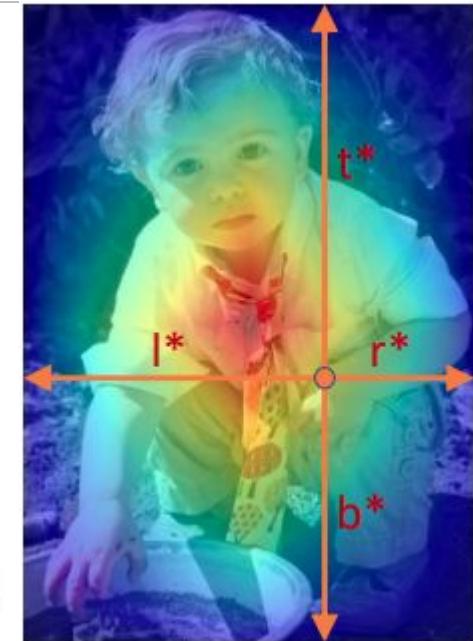
Функция потерь

$$L(\{\mathbf{p}_{x,y}\}, \{\mathbf{t}_{x,y}\}) = \frac{1}{N_{\text{pos}}} \sum_{x,y} L_{\text{cls}}(\mathbf{p}_{x,y}, c_{x,y}^*) + \frac{\lambda}{N_{\text{pos}}} \sum_{x,y} \mathbb{1}_{\{c_{x,y}^* > 0\}} L_{\text{reg}}(\mathbf{t}_{x,y}, \mathbf{t}_{x,y}^*),$$

- Focal loss для классификации
- IoU loss для регрессии

Center-ness

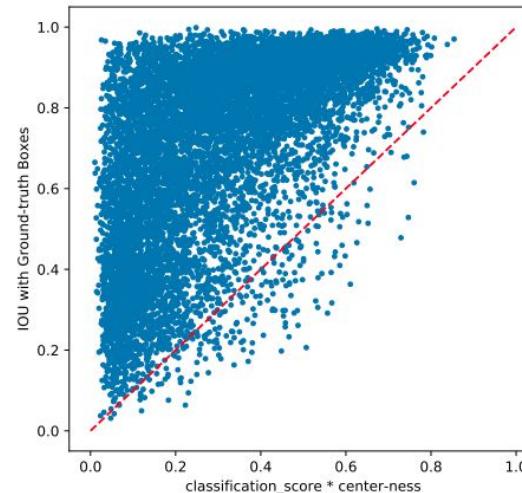
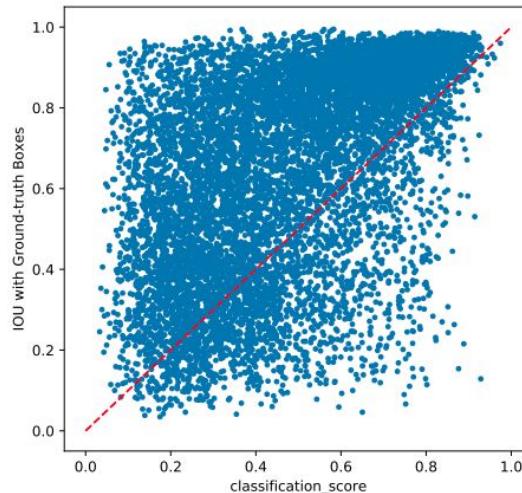
- Точки на краях объекта дают bounding box'ы плохого качества
- Дополнительная величина для предсказания
- Влияет на степень уверенности потенциальной принадлежности к объекту



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

Center-ness

	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
None	33.5	52.6	35.2	20.8	38.5	42.6
center-ness [†]	33.5	52.4	35.1	20.8	37.8	42.8
center-ness	37.1	55.9	39.8	21.3	41.0	47.8



Эксперименты

Method	# samples	AR ¹⁰⁰	AR ^{1k}
RPN w/ FPN & GN (ReImpl.)	~200K	44.7	56.9
FCOS w/ GN w/o center-ness	~66K	48.0	59.3
FCOS w/ GN	~66K	52.8	60.3

Method	Backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
Two-stage methods:							
Faster R-CNN w/ FPN [14]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [11]	Inception-ResNet-v2 [27]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w/ TDM [25]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
One-stage methods:							
YOLOv2 [22]	DarkNet-19 [22]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [18]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [5]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [15]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [13]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
FSAF [34]	ResNeXt-64x4d-101-FPN	42.9	63.8	46.3	26.6	46.2	52.7
FCOS	ResNet-101-FPN	41.5	60.7	45.0	24.4	44.8	51.6
FCOS	HRNet-W32-51 [26]	42.0	60.4	45.3	25.4	45.0	51.0
FCOS	ResNeXt-32x8d-101-FPN	42.7	62.2	46.1	26.0	45.6	52.6
FCOS	ResNeXt-64x4d-101-FPN	43.2	62.8	46.6	26.5	46.2	53.3
FCOS w/ improvements	ResNeXt-64x4d-101-FPN	44.7	64.1	48.4	27.6	47.5	55.6

Summary

- Anchor-free подход
- Разноуровневые карты признаков по типу FPN
- Center-ness для улучшения

EfficientDet

EVOCARGO



girafe
ai

Мотивация

Необходимо учитывать в современных моделях по ОД:

- Производительность модели
- Качество результата
- Ограничения на ресурсы

Запрос: построить масштабируемую архитектуру с высокой точностью и производительностью при разных ограничениях на ресурсы



EfficientDet

(Google Research, Brain Team) 2019.

Это one-stage детектор.

Состоит из:

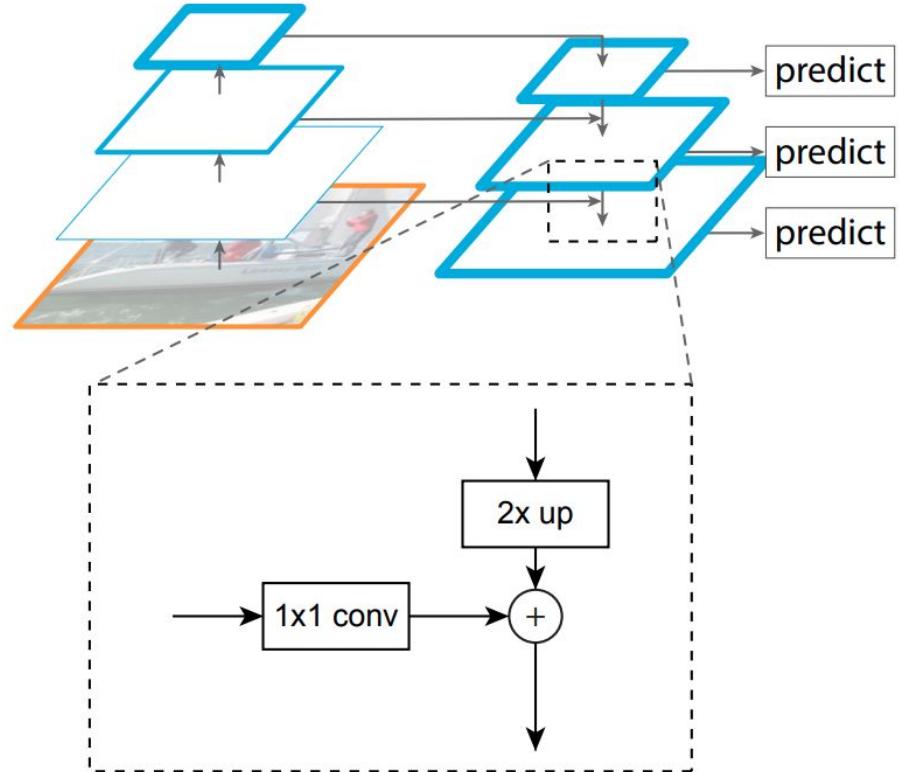
- Backbone
- Feature network
- Box/Class prediction



FPN

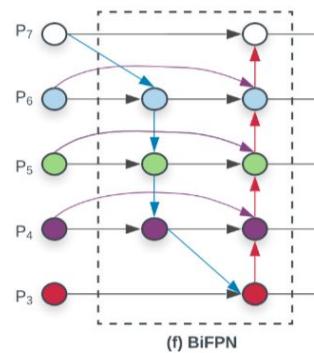
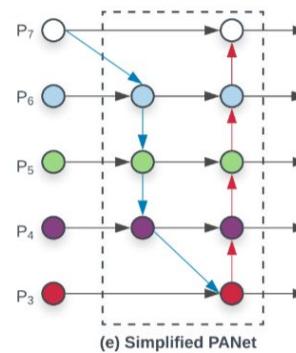
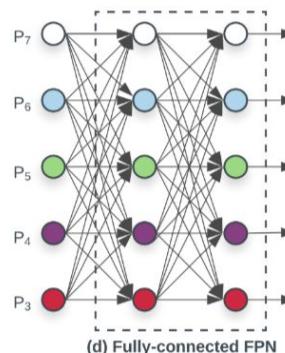
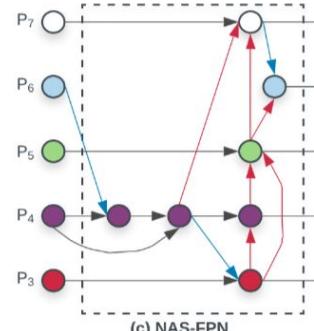
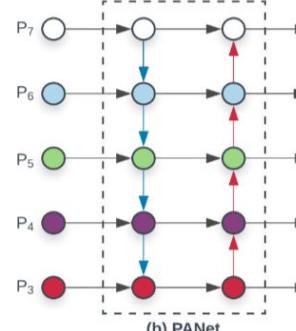
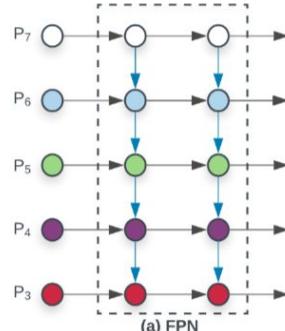
Feature Pyramid Networks (2016)

- Top-down архитектура
- Комбинация признаков с карт разного разрешения



Feature network

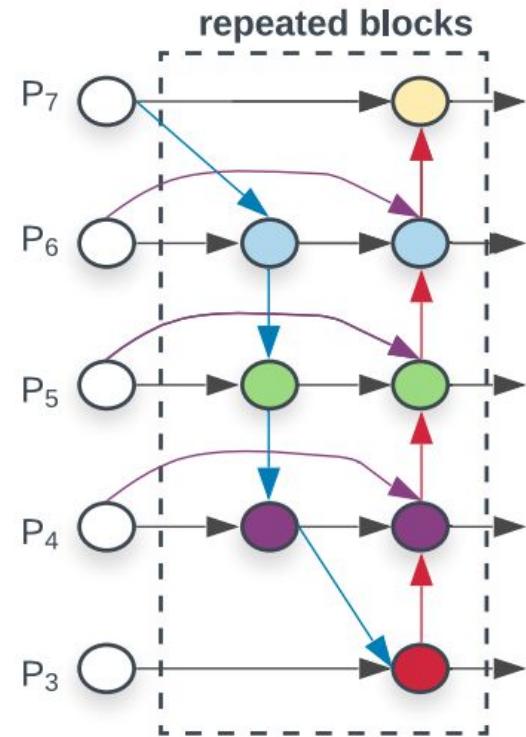
Используется идея Multi-scale feature fusion



BiFPN

- Убираем вершины с одним входом
- Прокидываем к выходам оригинальные слои
- Повторяется несколько раз

	AP	Parameters	FLOPs
ResNet50 + FPN	37.0	34M	97B
EfficientNet-B3 + FPN	40.3	21M	75B
EfficientNet-B3 + BiFPN	44.4	12M	24B



BiFPN

- Unbounded Fusion

$$O = \sum_i w_i \cdot I_i$$

- Softmax-based Fusion

$$O = \sum_i \frac{e^{w_i}}{\sum_j e^{w_j}} \cdot I_i$$

- Fast Normalized Fusion

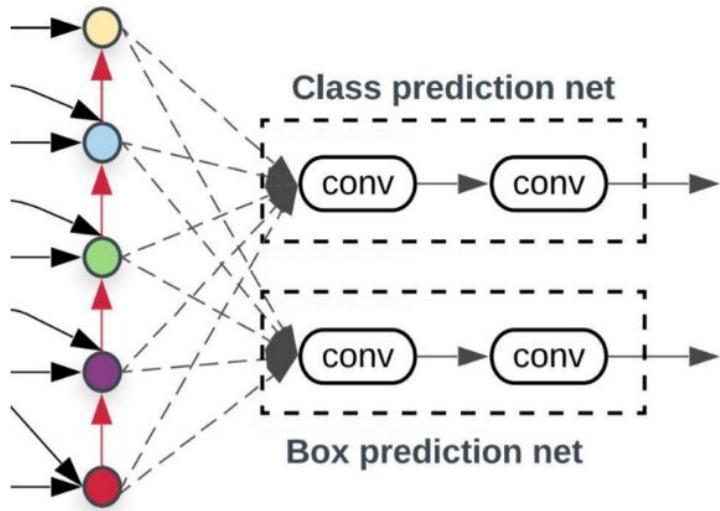
$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i$$

Model	Softmax Fusion AP	Fast Fusion AP (delta)	Speedup
Model1	33.96	33.85 (-0.11)	1.28x
Model2	43.78	43.77 (-0.01)	1.26x
Model3	48.79	48.74 (-0.05)	1.31x

	AP	#Params ratio	#FLOPs ratio
Repeated top-down FPN	42.29	1.0x	1.0x
Repeated FPN+PANet	44.08	1.0x	1.0x
NAS-FPN	43.16	0.71x	0.72x
Fully-Connected FPN	43.06	1.24x	1.21x
BiFPN (w/o weighted)	43.94	0.88x	0.67x
BiFPN (w/ weighted)	44.39	0.88x	0.68x

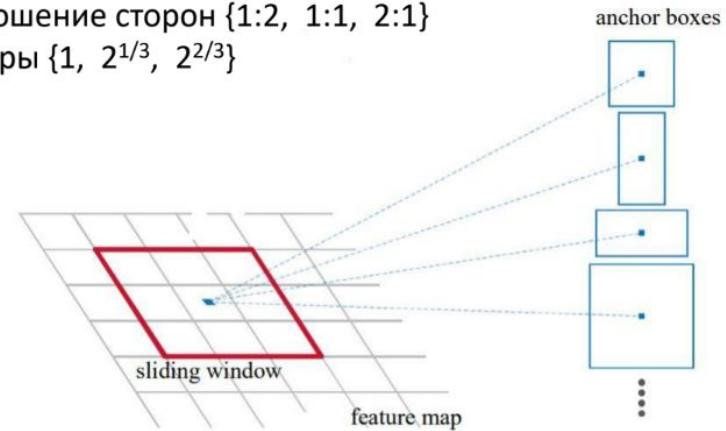


Box/Class prediction

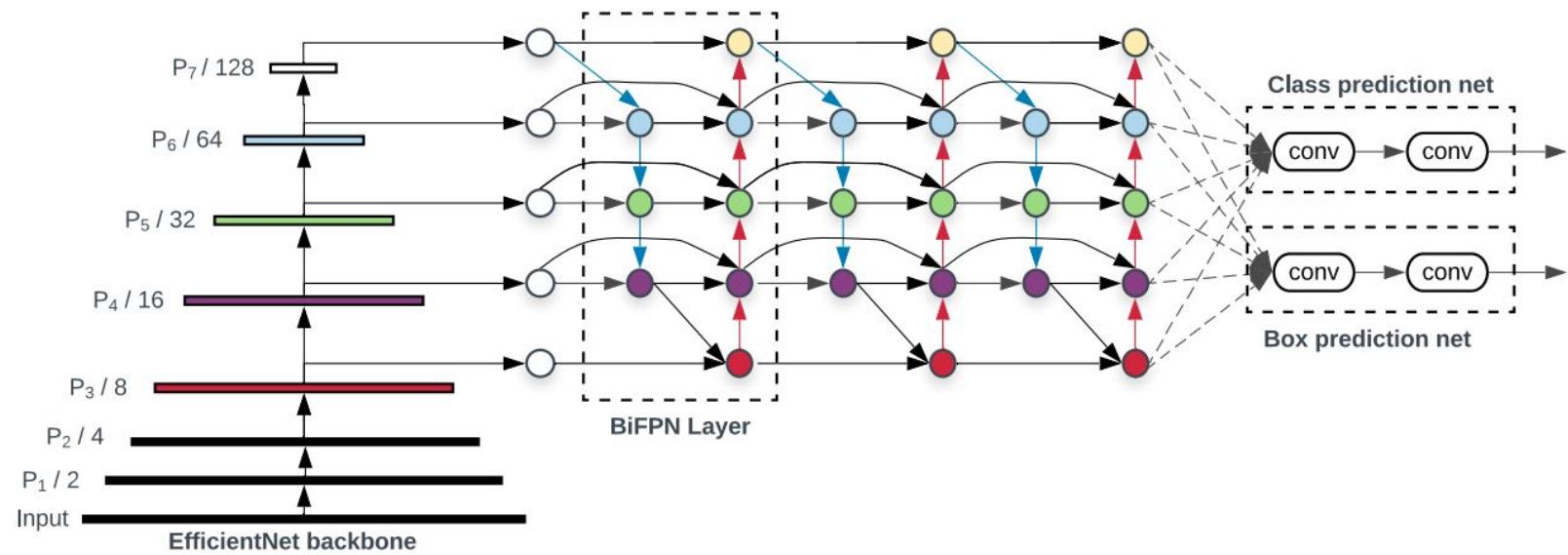


Anchor box:

- соотношение сторон {1:2, 1:1, 2:1}
- Размеры {1, $2^{1/3}$, $2^{2/3}$ }



Архитектура



Детали

- Loss: $L(p, y, t^y, v) = FL(p, y) + \lambda[y \geq 1]L_{reg}(t^y, v)$

где $\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$

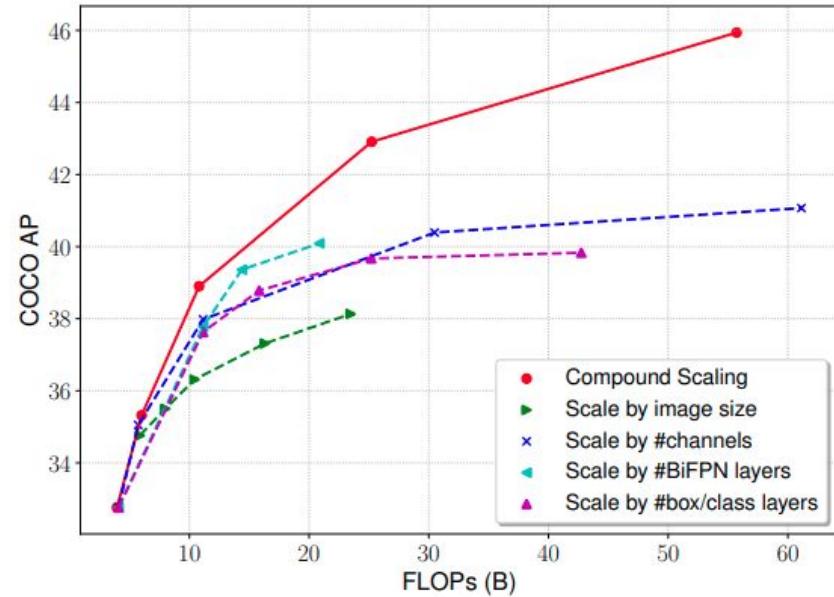
$$L_{reg}(t^y, v) = \sum_{i=1}^4 \text{smooth}_{L_1}(t_i^y - v_i)$$

- Датасет: COCO (#330000 картинок)



Compound Scaling

Неразумно изменять только
отдельные части



Compound Scaling

- Backbone (сама EfficientNet и разрешение картинки): $R_{input} = 512 + \phi \cdot 128$
- BiFPN: $W_{bifpn} = 64 \cdot (1.35^\phi)$, $D_{bifpn} = 3 + \phi$
- Box/class prediction network: $D_{box} = D_{class} = 3 + \lfloor \phi / 3 \rfloor$



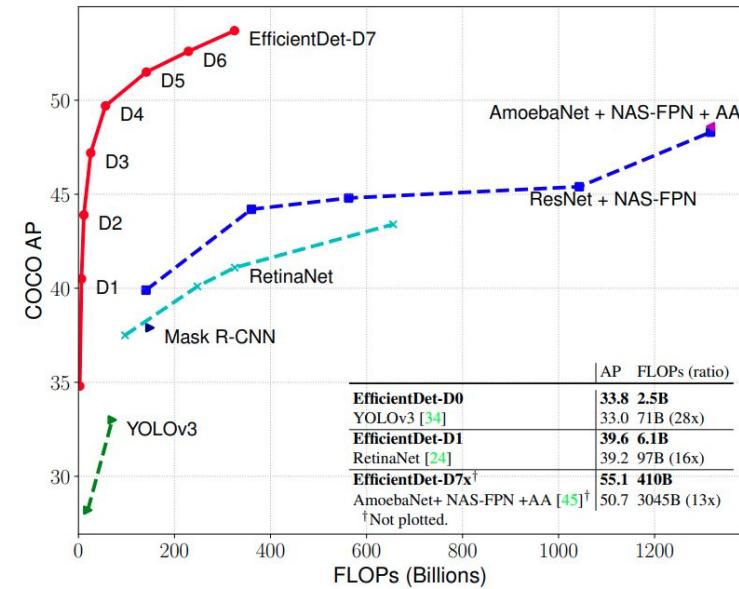
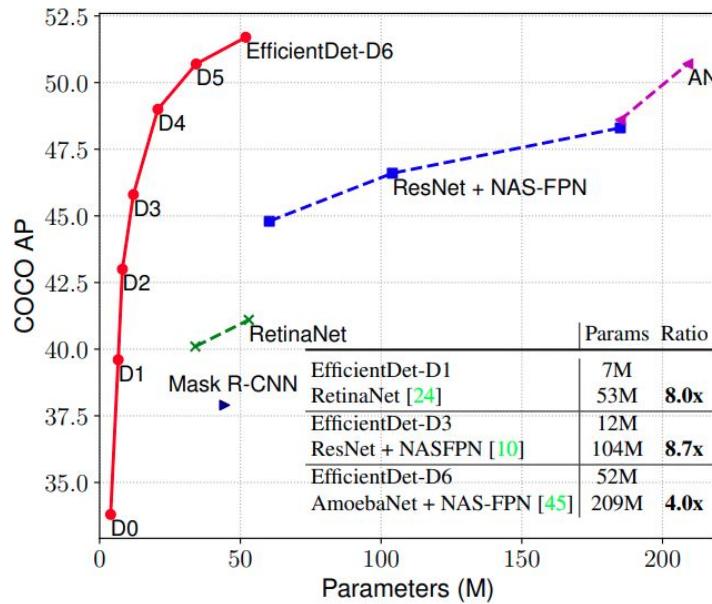
Compound Scaling

Больше 6 нет смысла

	Input size	Backbone Network	BiFPN		Box/class
	R_{input}		#channels W_{bifpn}	#layers D_{bifpn}	#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
D7 ($\phi = 7$)	1536	B6	384	8	5
D7x	1536	B7	384	8	5



Сравнение



Summary

- EfficientNet в качестве backbone
- Оригинальный BiFPN для построения карт признаков
- Эффективно масштабируемая сеть

