Focal Loss for Dense Object Detection. RetinaNet

Пудяков Ярослав 161

НИУ ВШЭ. НИС Машинное обучение и приложения (2019)

Содержание

1. Постановка задачи
2. Существующие подходы
3. Мотивация работы
4. Focall Loss
5. RetinaNet
6. Результаты
7. Выводы
8. Используемые источники

Постановка задачи Object Detection

• Для каждого объекта на изображении определить класс и выделить соответствующий ему bounding box.

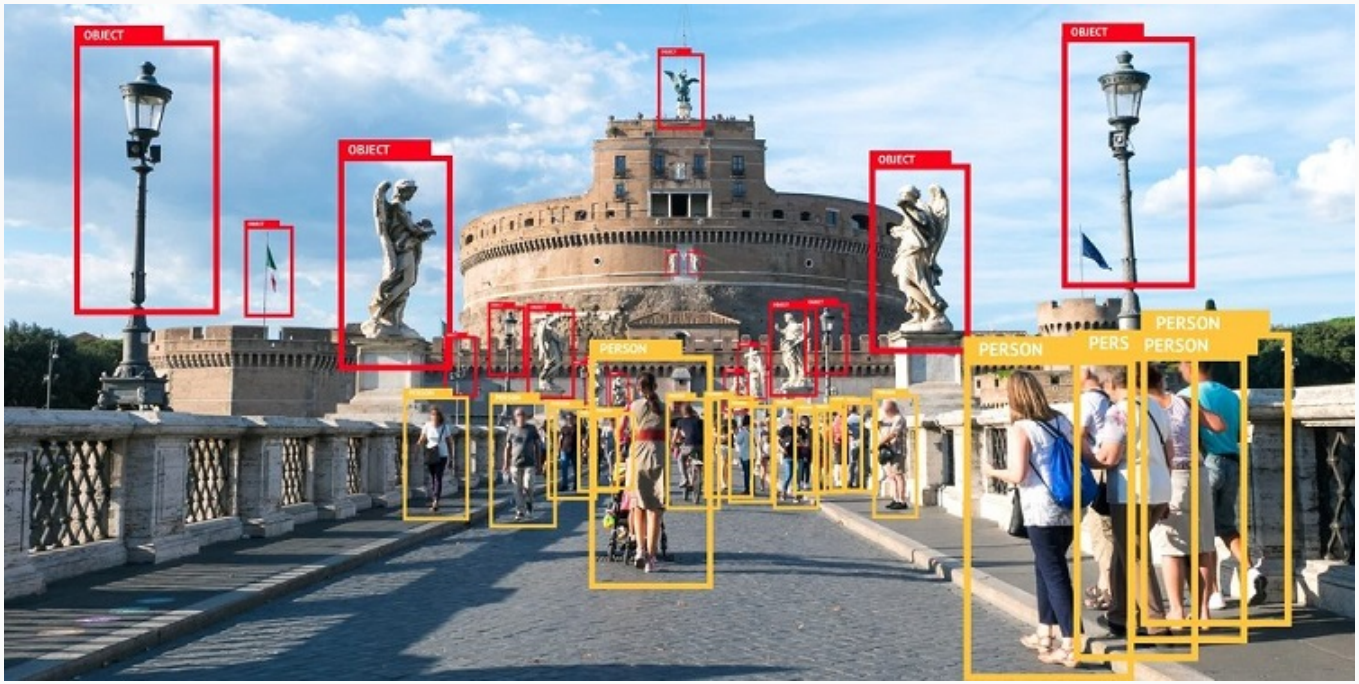


Рис. 1: Object Detection
(A Closer Look at Object Detection, Recognition and Tracking, Grevelink, Evelyn, December 18, 2017, software.intel.com)

Существующие подходы

- One-stages Detectors
  - *OverFeat, YOLO, SSD*
  - Генерируют плотную выборку возможных областей $\approx 10^5$ и напрямую пытается найти в них объекты

  $+$ Простая и быстрая модель.

  $-$ Результаты на 10% - 40% ниже, чем у two-stages detectors.
- Two-stages Detectors
  - *R-CNN, Faster-R-CNN*
  - Модель генерирует возможные области. Далее отсекаются изображения, где с большо̀й вероятностью не будет объекта, остаются только наиболее потенциальные кандидаты $\approx$ 1000 - 2000

  $+$ Результативность

  $-$ Время работы в разы больше, чем у one-stage детекторов.

Мотивация

- Можно ли добиться качества сравнимого с two-stages детекторами на one-stage детекторе?
- Основная проблема - class imbalance

  1. Выделяются тысячи областей-кандидатов, но лишь немногие содержат объекты
  2. В сумме, огромное множество негативных примеров (не содержащих объекты), могут сильно ухудшать модель.

Focal Loss

- Решением является использование специальной функции ошибки - Focal Loss.
- По аналогии с Huber Loss, направленной на заглушение влияния outliers, Focal Loss направлена на уменьшие веса inliers (легких отрицательных примеров)
- Focal Loss концетрирует обучение на множестве редких и сложных примеров.

Cross Entropy (CE)

Focal Loss основан на улучшении стандартной функции кросс-энтропии:

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases}$$

p - вероятность класса $y = 1$, предсказанная моделью.

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases}$$

Перепишем функции ошибки как:

$$\mathrm{CE}(p, y) = \mathrm{CE}\left(p_{\mathrm{t}}\right) = -\log\left(p_{\mathrm{t}}\right)$$

Balanced Cross Entropy

Для борьбы с несбалансированными классами, обычно вводится весовой фактор $\alpha \in [0, 1]$ для $y = 1$, и $1 - \alpha$ для $y = 0$. На практике он устанавливается равным обратной частоте класса $y = 1$, либо подбирается на кроссвалидации.

Balanced-CE:

$$\mathrm{CE}\left(p_{\mathrm{t}}\right) = -\alpha_{\mathrm{t}} \log\left(p_{\mathrm{t}}\right)$$

Focal Loss Definition

Balanced-CE концентрируется на классе изображения, учитывая в равной степени влияние каждого, но совсем не учитывает сложность примера - легкий ли он для детектирования, или наоборот.

Чтобы учитывать сложность примера, вводится modulating factor $(1 - p_t)^\gamma$, $\gamma \geq 0$

Focal Loss:

$$\mathrm{FL}\left(p_\mathrm{t}\right) = -\left(1 - p_\mathrm{t}\right)^\gamma \log\left(p_\mathrm{t}\right)$$

Свойства Focal Loss

$$\mathrm{CE}\left(p_{\mathrm{t}}\right) = -\log\left(p_{\mathrm{t}}\right)$$

$$\mathrm{FL}\left(p_{\mathrm{t}}\right) = -\left(1-p_{\mathrm{t}}\right)^{\gamma}\log\left(p_{\mathrm{t}}\right)$$

Focal Loss обладает следующими свойствами:

- Когда пример неправильно классифицирован и $p_t \to 0$, то $(1-p_t)^{\gamma} \approx 1$ и ошибка схожа с $CE(p_t)$.
- Когда степень уверенности в примере очень большая $p_t \to 1$, то $(1-p_t)^{\gamma} \approx 0$ и ошибка на объекте в разы меньше, чем в случае с $CE(p_t)$.
- Параметр γ регулирует степень значимости легко классифицируемых примеров. Чем больше γ, тем меньше вклад легких примеров в обучение.

Таким образом, Focal Loss заостряет внимание на труднoклассифицируемых примерах.

Balanced Focal Loss. Сравнение функций ошибок.

На практике обычно используется α-balanced FL.

$$\mathrm{FL}\left(p_{\mathrm{t}}\right) = -\alpha_{\mathrm{t}}\left(1-p_{\mathrm{t}}\right)^{\gamma}\log\left(p_{\mathrm{t}}\right)$$

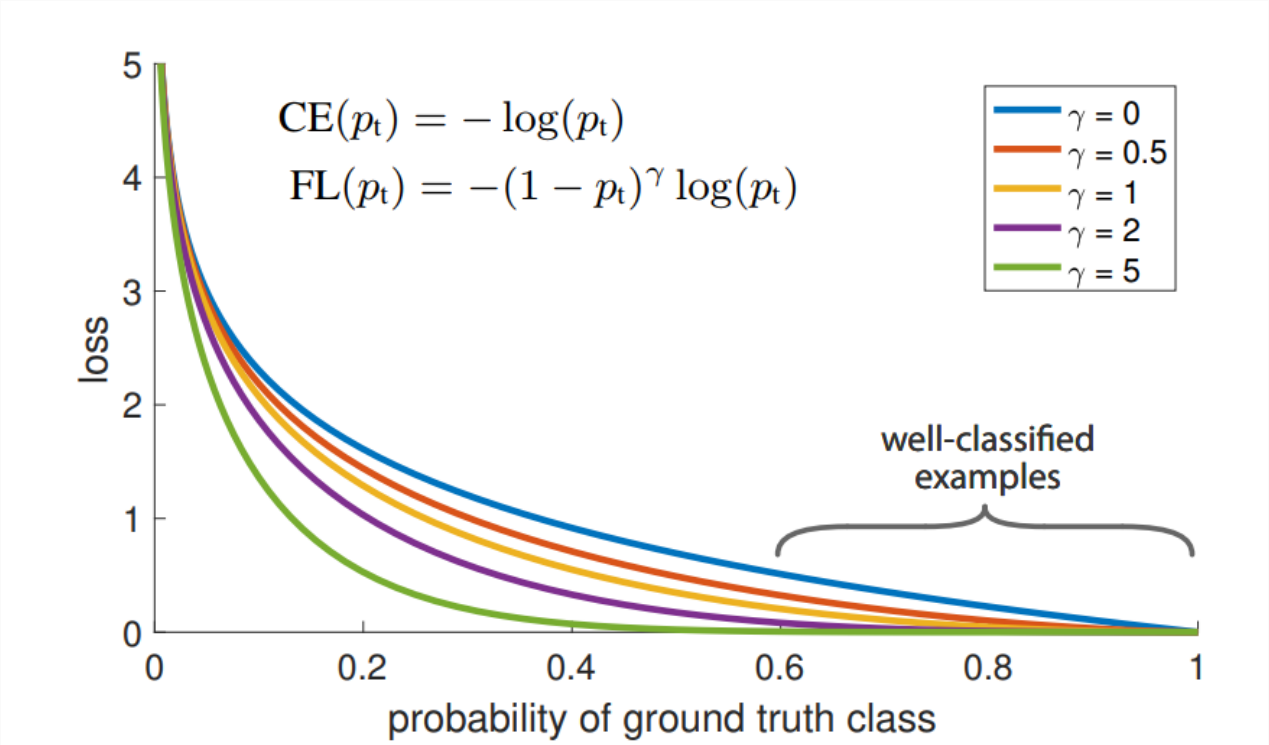


Рис. 2: Сравнение ошибок на объекте для разной степени уверенности модели, при различном параметре γ
(Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and PiotrDoll ar. Focal Loss for Dense Object Detection, 2017)

RetinaNet

RetinaNet - one-stage детектор, обучающийся при помощи Focal Loss.

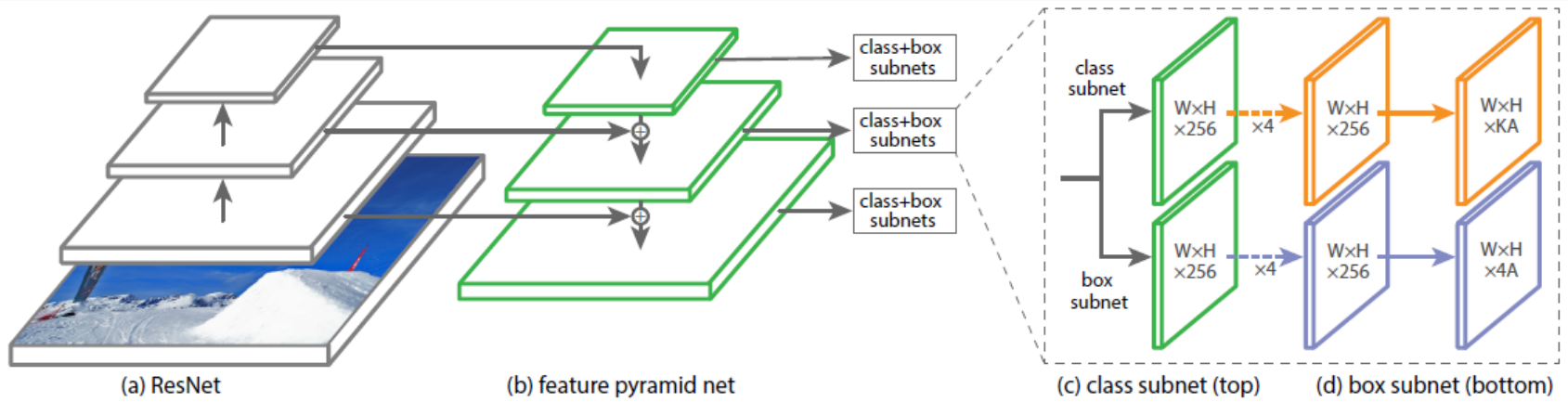
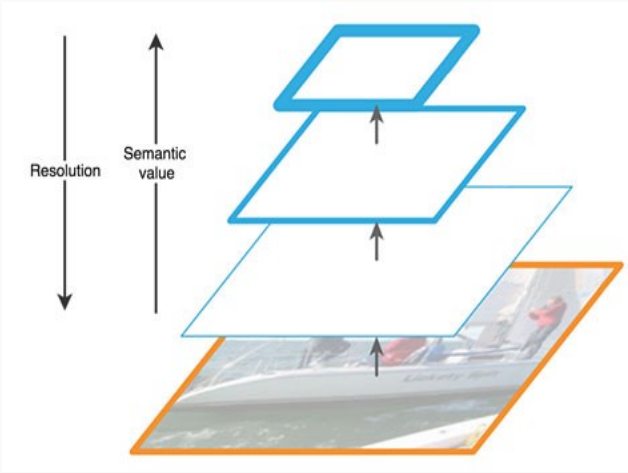


Рис. 3: Архитектура сети RetinaNet.
(Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and PiotrDoll ar. Focal Loss for Dense Object Detection, 2017)

Feature Pyramid Networks (FPN)



- Каждый уровень пирамиды используется для детекции
- С повышением уровня уменьшается разрешение, и растет семантическое значение
- На всех уровнях количество каналов $C = 256$
- m уровень, имеет разрешение в 2^m меньше, чем исходное изображение

Feature Pyramid Networks (FPN)

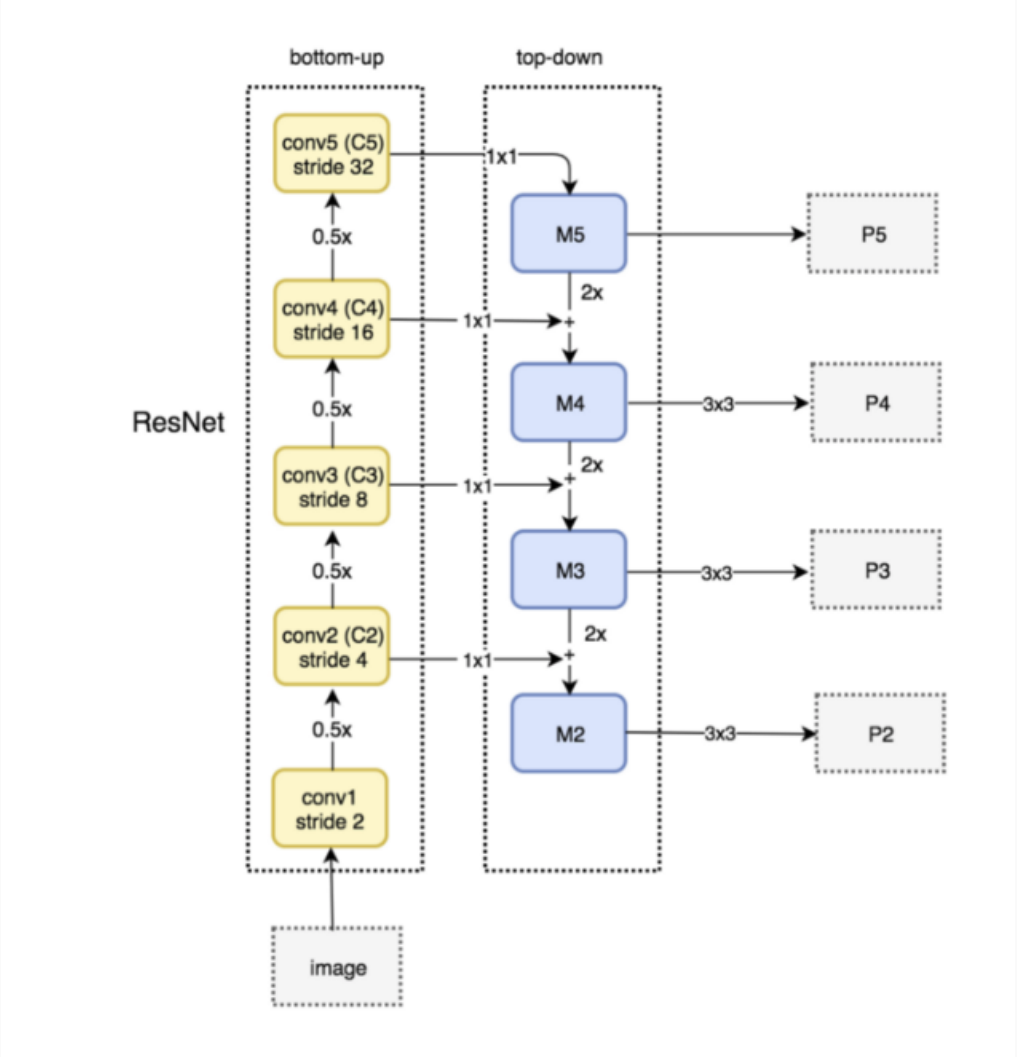


Рис. 4: FPN схема. (Review: FPN - Feature Pyramid Network (Object Detection), Sik-Ho Tsang, towardsdatascience.com, 2019)

Anchors

После извлечения карты признаков, строятся области, которые могут содержать объект.

- Области имеют площади от 32^2 до 512^2 на уровнях от P_7 до P_3 соответсвенно.
- На каждом уровне строятся прямоугольные области с соотношением сторон 1:1, 1:2, 2:1.
- Для каждого соотношения сторон рассматриваются три различных масштаба $2^0, 2^{1/3}, 2^{2/3}$
- Таким образом, на каждом уровне выделяются по 9 различных областей
- Если IoU региона с реальным bounding box больше 0.5, то мы считаем, что регион задетекли объект. Меньше 0.4 - считаем, что объекта нет и это фон. Если $0.4 \leq \text{IoU} < 0.5$, то игнорируем регион при обучении.

Anchors. Пример.

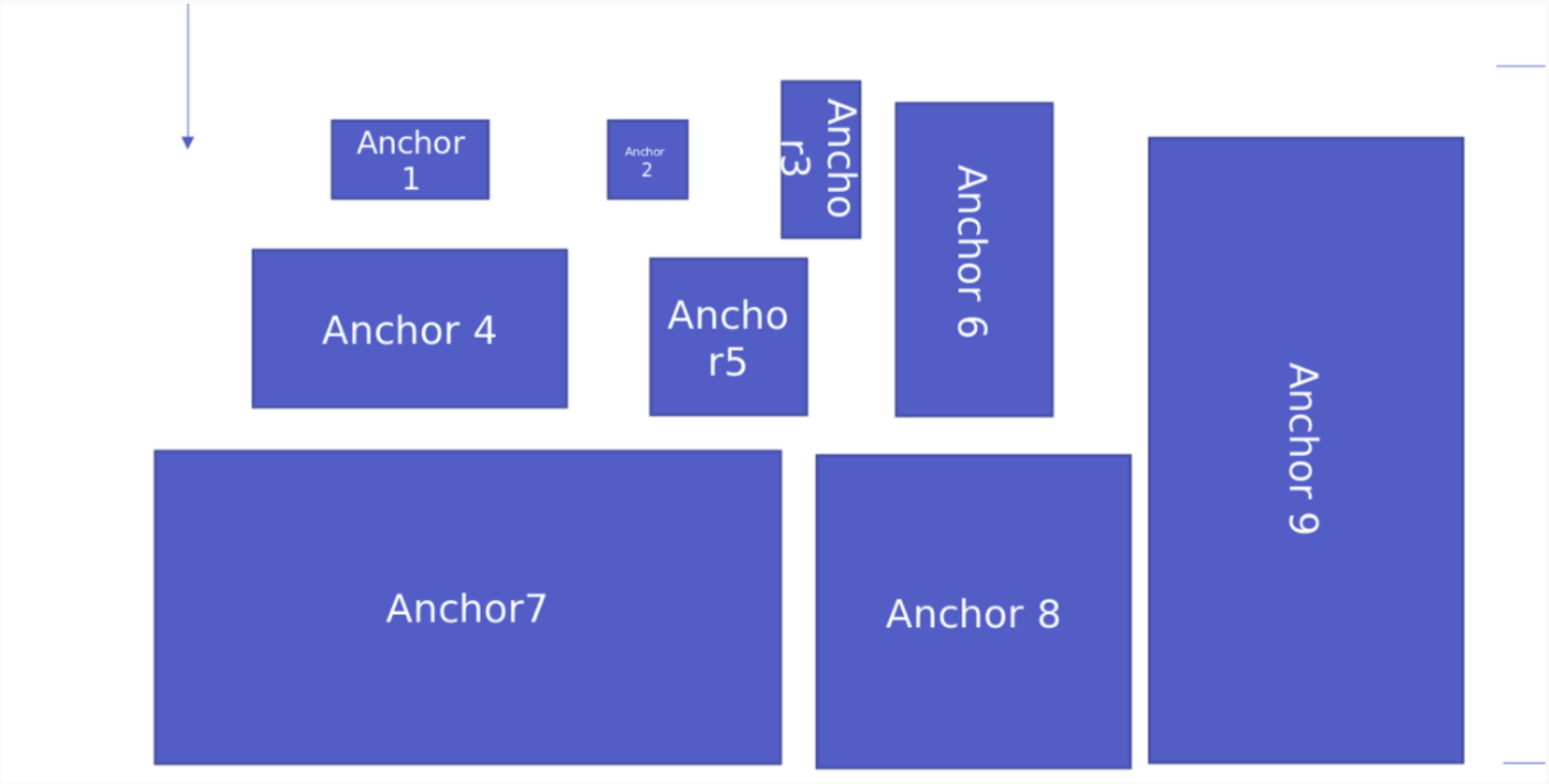


Рис. 5: Различные генерируемые регионы. (The intuition behind RetinaNet, Prakash Jay, medium.com, 2018)

Classification Subnet

- Данная предсказывает вероятность нахождения класса в данном регионе. Для каждого сгенерированного региона считаются вероятности для каждого из K классовPrakash Jay, medium.com, 2018
- Сеть представляет из себя маленькую Fully Convolutional Network. К каждому уровню пирамиды прикреплена своя FCN.
- Архитектура сети: 4 блока из ($3x3@C$+ReLU), после этого одна свертка ($3x3xKA$) и примение к результату сигмоиды $\sigma(\cdot)$

Box Regression Subnet

- Сеть работает параллельно Classification Subnet
- Сеть представляет из себя маленькую Fully Convolutional Network. К каждому уровню пирамиды прикреплена своя FCN.
- Архитектура сети: идентична Classification Subnet, исключая лишь то, что на выходе она выдает местоположения объектов, которых $4A$

Итоговая обработка

• Для увеличения скорости работы сеть обрабатывает не более чем top-1000 предсказаний регионов, которые наиболее вероятны, после уверенности модели более чем на 0.05.

• Лучшие прогнозы со всех уровней объединяются, и для получения окончательных обнаружений применяется немаксимальное подавление (NMS) с порогом 0,5.

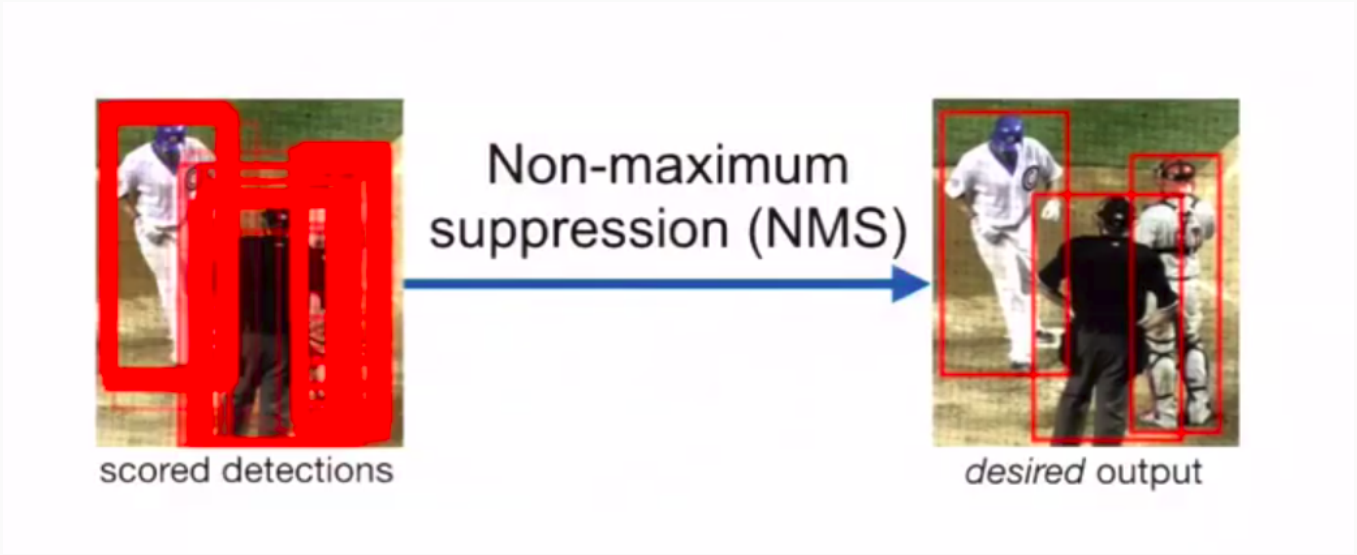


Рис. 6: Non-maximum suppression.
(Non-Maximum Suppression for Object Detection in Python, www.pyimagesearch.com, 2014)

Обучение

- Таким образом, во время обучения общий loss изображения вычисляется как сумма focal loss для всех сгенерированных регионов ($\approx 10^5$), нормализуя по количеству регионов, назначенному groundtruth bb
- Обучаем при помощи SGD

Результаты

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)

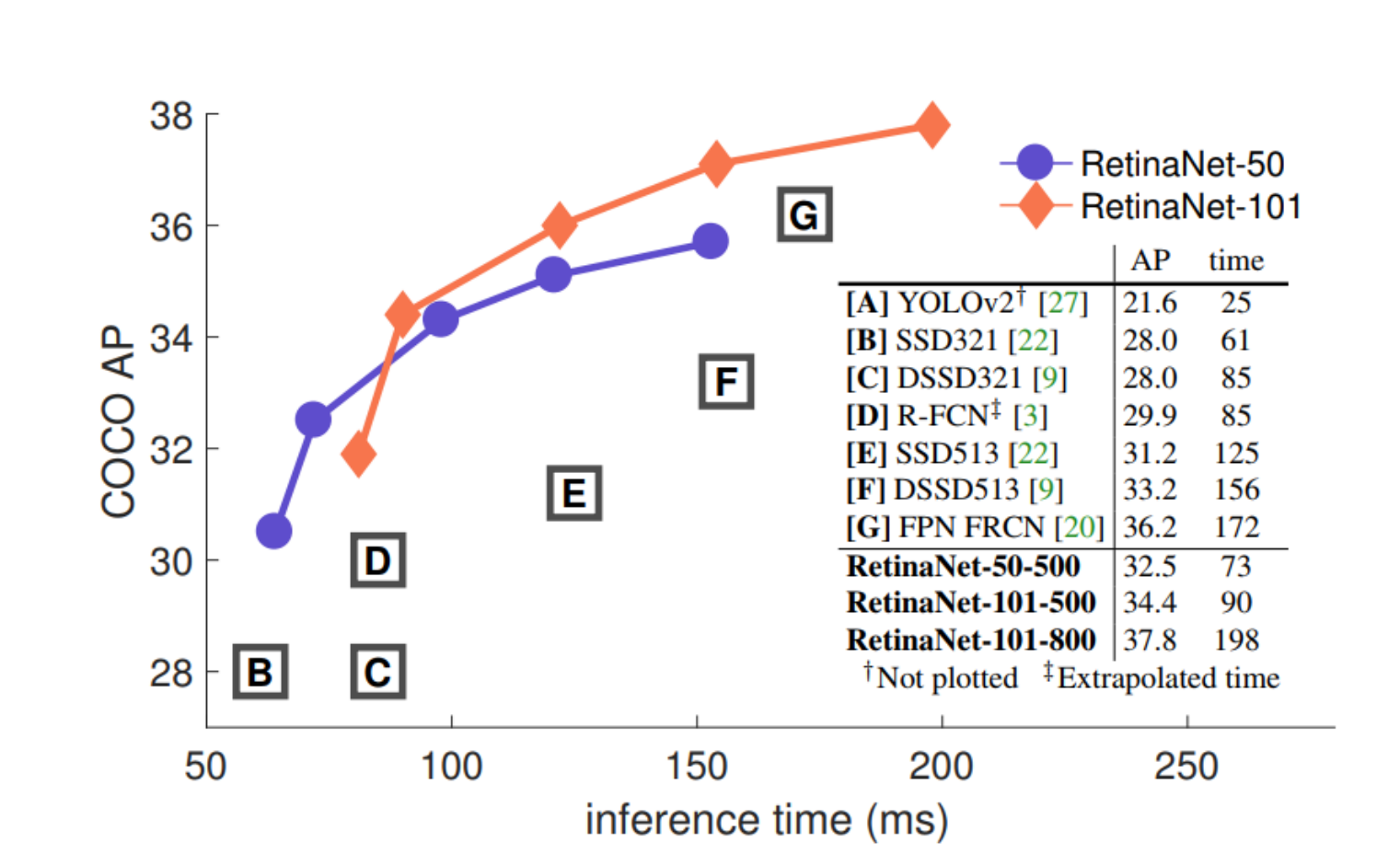
Рис. 7: Подбор параметром γ и α.
(Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and PiotrDoll ar. Focal Loss for Dense Object Detection, 2017)

Результаты

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 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
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

Рис. 8: Сравнение различных нейронных сетей.
(Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and PiotrDoll ar. Focal Loss for Dense Object Detection, 2017)

Результаты



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	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

Рис. 9: Сравнение различных нейронных сетей.
(Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and PiotrDoll ar. Focal Loss for Dense Object Detection, 2017)

Выводы

1. При помощи Focal Loss можно учитывать сложность примеров, а также производить балансированность по классам.
2. RetinaNet one-stage детектор обученный при помощи Focal loss, превосходит все известные two-stage детекторы.
3. RetinaNet не только получает превосходные результаты, но и довольно быстро отрабатывает. От 73-90с до 200с на изображениях из COCO, в зависимости от backbone в FPN

Используемые источники

1. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and PiotrDoll ar. Focal Loss for Dense Object Detection.arXiv e-prints, pagearXiv:1708.02002, Aug 2017

2. The intuition behind RetinaNet, Prakash Jay, medium.com, 2018

3. Review: FPN - Feature Pyramid Network (Object Detection), Sik-Ho Tsang, towardsdatascience.com, 2019