УДК 004.8, 004.93

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ ОБНАРУЖЕНИЯ ОБЪЕКТОВ СОВРЕМЕННЫХ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Колбасов С.Ю., Орлов Ю.К.
Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
E-mail: kolbasovsergey02@gmail.com

Аннотация

Колбасов С.Ю., Орлов Ю.К. Сравнение эффективности современных сверточных нейронных сетей для задачи обнаружения объектов. *Данная работа посвящена сравнительному анализу точности и скорости обнаружения объектов современных сверточных нейронных сетей: Faster R-CNN, SSD, YOLO. Результаты показывают, что Faster R-CNN и SSD имеют высокую точность и превосходят YOLO, однако YOLO и SSD более быстрые и могут быть использованы для обнаружения объектов в реальном времени.*

Ключевые слова: *обнаружение объектов, сверточная нейронная сеть.*

Abstract

Kolbasov S.Y., Orlov Y.K. Comparison of the effectiveness of modern convolutional neural networks for the task of object detection. *This work is devoted to a comparative analysis of the accuracy and speed of object detection of modern convolutional neural networks: Faster R-CNN, SSD, YOLO. The results show that Faster R-CNN and SSD are highly accurate and superior to YOLO, however YOLO and SSD are faster and can be used for real-time object detection.*

Keywords: *object detection, convolutional neural network.*

Введение. Люди смотрят на изображение и мгновенно узнают, какие объекты на нем находятся, где они находятся и как они взаимодействуют. Человеческая зрительная система быстрая и точная, что позволяет нам выполнять сложные задачи, например, управлять автомобилем, не задумываясь. Быстрые, точные алгоритмы обнаружения объектов позволили бы компьютерам управлять машинами без специализированных датчиков, вспомогательным устройствам передавать информацию о сцене в реальном времени пользователям и открыли бы потенциал для универсальных, отзывчивых роботизированных систем.

Современные системы обнаружения изменяют классификаторы для выполнения обнаружения. Чтобы обнаружить объект, такие системы используют классификатор для этого объекта и оценивают его в различных местах и масштабах на тестовом изображении. На сегодняшний день в области обнаружения объектов доминируют сверточные нейронные сети. Однако для успешного обнаружения недостаточно взять любую из сетей и надеяться на хороший результат, так как все сети отличаются различными техниками и подходами, архитектурами, но, самое главное, своей эффективностью. Поэтому имеет смысл провести сравнительный анализ современных сверточных нейронных сетей для обнаружения объектов.

Цель статьи. Обучение и последующее сравнение эффективности обнаружения объектов современных сверточных нейронных сетей, а именно: Faster R-CNN, SSD, YOLO.

Обучение Faster R-CNN. Faster R-CNN, по сути, состоит из двух модулей [1]. Первый модуль представляет собой глубокую сверточную сеть, которая предлагает регионы (англ. Region Proposal Network, RPN), и второй модуль – детектор Fast R-CNN [2], который использует предложенные регионы. Сеть RPN использует карту признаков последнего сверточного слоя, чтобы предсказывать регионы. В каждом месте карты признаков сеть

одновременно предсказывает несколько предложений регионов, где число максимально возможных предложений для каждого места обозначается как *k*. *K* предложений параметризованы относительно *k* ссылочных блоков, которые называются якорями. Якорь центрирован на скользящем окне, и связан с масштабом и соотношением сторон. Используется 3 масштаба и 3 соотношения сторон, что дает *k* = 9 якорей на каждой позиции скольжения.

Во время работы RPN каждому якорю присваивается метка двоичного класса (является объектом или нет). Положительная метка присваивается двум видам якорей:

– якорь (якоря) с наивысшей оценкой перекрытия по отношению к истинному значению ограничивающей рамки;

– якорь, который имеет оценку перекрытия выше 0,7 с любым истинным значением.

Отрицательная метка (не является объектом) присваивается якорю, если его оценка перекрытия ниже 0,3 для любого истинного значения. Остальные якоря не вносят вклад в обучение.

Функция потерь для изображения определяется как:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i). \quad (1)$$

В приведенной формуле (1) i – это индекс якоря, а p_i – прогнозируемая вероятность того, что якорь является объектом. p_i^* – метка истинности (равна 1, если якорь положительный, и 0, если отрицательный). t_i – это вектор, представляющий 4 параметризованные координаты предсказанной ограничивающей рамки, а t_i^* – это метка истинности связанная с положительным якорем. Потеря классификации L_{cls} – это логарифмическая функция потерь по двум классам (объект или не объект). L_{reg} – это регрессионная функция потерь. Выражение $p_i^* L_{reg}$ означает, что регрессионная функция потерь активируется только для положительных якорей. Два слагаемых нормализуются при помощи N_{cls} и N_{reg} и взвешиваются по балансирующему параметру λ. По умолчанию, в качестве N_{cls} используется размер мини-пакета (часть изображений, переданных в нейронную сеть), а в качестве N_{reg} используется количество якорей (обычно примерно 2400). По умолчанию параметр λ = 10, что делает оба члена *cls* и *reg* примерно одинаково взвешенными.

Для обучения RPN используется метод обратного распространения ошибки и стохастического градиентного спуска (англ. Stochastic gradient descent, SGD). Каждый мини-пакет получается из одного изображения, которое содержит много положительных и отрицательных якорей. Случайным образом выбирается 256 якорей на изображении, чтобы вычислить функцию потерь для мини-пакета, где выборочные положительные и отрицательные якоря имеют соотношение до 1:1. Если на изображении меньше 128 положительных якорей, то они дополняются отрицательными. Все веса слоев инициализируются при помощи распределения Гаусса с математическим ожиданием 0 и стандартным отклонением 0,01. Параметр скорости обучения равен 0,0001 для мини-пакетов по 20 тысяч, инерцией (англ. momentum) 0,9, и затуханием весов (англ. weight decay) 0,0005.

Обучение SSD. Ключевое различие между обучением SSD [3] и обучением типичного детектора, использующего предложения регионов, заключается в том, что информацию об истинном значении (метке) необходимо назначать конкретным выходам в фиксированном наборе выходов детектора. Как только это назначение определено, функция потерь и обратное распространение применяются непрерывно.

Общая функция потери представляет собой взвешенную сумму потерь локализации (loc) и потери уверенности (conf):

$$L(x,c,l,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\right). \quad (2)$$

В формуле (2) N – это количество совпавших рамок по умолчанию. Если N = 0, то потеря будет равна 0. Потеря локализации (L_{loc}) представляет собой функцию потерь L1 между параметрами прогнозируемой рамки (l) и истинной рамки (g). Потеря уверенности (L_{conf}) представляет собой функцию потерь softmax по классовым уверенностям (c).

Во время обучения модели используется метод SGD с параметром скорости обучения 0,001, инерцией равной 0,9, затуханием весов 0,0005 и размером пакета 32.

Обучение YOLO. Результатом работы модели YOLO [4] является предсказание как вероятности класса, так и координаты ограничивающей рамки. Ширина и высота рамки нормализуется по ширине и высоте изображения таким образом, чтобы они находились между 0 и 1. Координаты рамки x и y параметризуются так, чтобы они были смещениями определенной ячейки сетки, чтобы они также были ограничены между 0 и 1. На выходе модели используется сумма квадратов ошибок, потому что ее легко оптимизировать. Однако авторы модели отмечают, что такой выбор может быть не самым удачным, так как сумма квадратов взвешивает ошибку локализации в равной степени с ошибкой классификации, которая может быть не идеальной. Кроме того, в каждом изображении многие ячейки сетки не содержат никаких объектов. Это подталкивает оценку «уверенности» этих ячеек к нулю, часто подавляя градиент от ячеек, которые содержат объекты. Это может привести к нестабильности модели, что приведет к тому, что обучение на ранних этапах расходится.

Чтобы исправить это, увеличивается потеря от предсказаний координат ограничивающей рамки и уменьшается потеря от предсказаний уверенности для рамок, которые не содержат объектов. Для этого используются два параметра, λ_{coord} и λ_{noobj}. По умолчанию λ_{coord} = 5 и λ_{noobj} = 0,5.

Для обучения используется следующая, состоящая из нескольких частей, функция потерь:

$$\begin{aligned}
&\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right] \\
&+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right] \\
&+\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2 \\
&+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2 \\
&+\sum_{i=0}^{S^2} 1_{i}^{obj}\sum_{c\ \epsilon\ classes}\left(p_i(c)-\hat{p}_i(c)\right)^2. \quad (3)
\end{aligned}$$

В формуле (3) 1_i^{obj} обозначает, появляется ли объект в ячейке i, а 1_{ij}^{obj} обозначает, что j-й предсказатель рамки в ячейке i «отвечает» за это предсказание. S является размером сетки, B представляет количество ограничивающих рамок в каждой ячейке сетки, C представляет вероятности классов. Центр ограничивающей рамки представлен в виде координат x и y относительно границ ячейки сетки, w и h представляют ширину и высоту ограничивающей рамки относительно всего изображения.

Во время обучения используется метод SGD, с инерцией равной 0,9, затуханием весов 0,0005 и размером пакета 64. Что касается параметра скорости обучения, то для первых эпох

он медленно повышается с 0,001 до 0,01. Авторы статьи заметили, что если начинать с высокой скоростью обучения, то модель часто расходится из-за нестабильных градиентов.

Набор данных. Для обучения нейронных сетей использовался набор данных PASCAL Visual Object Classes (VOC). Данный набор данных предназначен для распознавания объектов из ряда классов визуальных объектов в реалистичных сценах (то есть не предварительно сегментированных объектах). Обычно VOC используется для трех основных соревнований распознавания объектов: классификация, обнаружение и сегментация. Этот набор данных используется для обучения с учителем, так как предоставляет обучающий набор помеченных изображений. Всего набор содержит двадцать классов объектов:

– человек: человек;

– животное: птица, кошка, корова, собака, лошадь, овца;

– транспортное средство: самолет, велосипед, лодка, автобус, автомобиль, мотоцикл, поезд;

– в помещении: бутылка, стул, обеденный стол, растение в горшке, диван, телевизор (монитор).

Набор данных VOC предоставляет данные для обучения и для тестирования. Набор разделен на 50% для обучения и 50% для тестирования. Распределение изображений и объектов по классам примерно одинаково по обучающим и тестовым наборам.

Данные для обучения состоят из набора изображений, каждое из которых имеет файл аннотации, содержащий координаты ограничивающей рамки и метку класса объекта для каждого объекта в одном из двадцати классов, представленных на изображении (несколько объектов из нескольких классов могут присутствовать на одном изображении). Всего обучающий набор содержит 5717 изображений (13609 объектов) для обучения и 5823 изображений (13841 объектов) для проверки.

Данные для тестирования предоставляются как набор изображений, не имеющий никакой информации о классах объектов и координатах ограничивающих рамок. Так как доступ к тестовому набору предоставляется по расписанию во время проведения соревнования, то для тестирования могут использоваться изображения для валидации.

Критерии сравнения. В качестве критерия для сравнения нейронных сетей обычно выступает точность (или уверенность). Однако данный критерий может вычисляться различными способами. В соревновании PASCAL Visual Object Classes Challenge и некоторых других используется метрика mAP (mean average precision), которая предназначена для определения того, насколько хорошо работает наша модель. Так как в данной работе используется набор данных из этого соревнования, то основным критерием для сравнения моделей будет точность в виде метрики mAP.

Данная метрика вычисляется по формуле:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i . \qquad (4)$$

В формуле (4) N представляет количество классов, а AP (average precision) – это ещё одна популярная метрика для измерения точности детекторов объектов [5]. То есть, по каждому из представленных классов в наборе данных вычисляется средняя точность, после чего вычисляется среднее этих точностей по всем классам.

Другим критерием для оценки является частота кадров. Этот критерий поможет определить, сколько кадров в секунду (англ. frames per second, FPS) может обрабатывать нейронная сеть.

Результаты. После обучения каждой нейронной сети на обучающем наборе данных их необходимо протестировать на «незнакомых» данных (тестовый набор), чтобы выяснить, насколько сеть научилась «обобщать» выученную информацию и обнаруживать объекты на новых изображениях.

По результатам тестирования модель Faster R-CNN показала точность в 73,2%, что является хорошим результатом для моделей обнаружения объектов. При этом, Faster R-CNN обрабатывает всего 7 кадров в секунду и поэтому не может быть использована для обнаружения объектов в реальном времени.

Модель SSD показала отличный результат в 74,3% mAP, превышая Faster R-CNN на 1.1%. Такая модель может эффективно использоваться для задач обнаружения объектов. SSD также достигла хорошего результата в скорости обработки кадров, обрабатывая 59 кадров в секунду, что делает SSD отличной сетью для обнаружения объектов на видео.

YOLO показала результат в 63,4% точности, что является наименьшим значением среди сравниваемых нейронных сетей. Однако, YOLO способна обрабатывать 45 кадров в секунду, что делает её более предпочтительной, чем Faster R-CNN для задач обнаружения в реальном времени.

Все полученные результаты сравнения нейронных сетей указаны в таблице 1.

Таблица 1. Результаты моделей на тестовом наборе Pascal VOC

Модель	mAP	FPS
Faster R-CNN	73,2	7
SSD	74,3	59
YOLO	63,4	45

Выводы. В данной работе был проведен сравнительный анализ эффективности обнаружения объектов современных сверточных нейронных сетей, используя набор данных VOC.

По результатам тестирования модели Faster R-CNN и SSD показали высокую точность, опережая YOLO примерно на 10%. Однако, YOLO имеет преимущество в скорости (45 FPS) и, также как и SSD (59 FPS), может использоваться для обнаружения объектов в реальном времени. Основным недостатком Faster R-CNN является скорость (7 FPS), то есть модель не может использоваться для обнаружения объектов в реальном времени.

На сегодняшний день можно сказать, что нейронные сети могут точно обнаруживать объекты, но каждая из них также имеет существенные минусы, например, YOLO плохо справляется с обнаружением маленьких объектов, а SSD затрачивает много времени на проход первых слоев. Поэтому выбор модели должен зависеть от решаемой задачи.

Литература

1. Shaoqing, R. Faster R-CNN: Towards Real-Time Object Detection with Regional Proposal Networks / R. Shaoqing, H. Kaiming, R. Girshick, J. Sun // IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, N 6, 2016. – P. 1137-1149.
2. Girshick, R. Fast R-CNN // 2015 IEEE International Conference on Computer Vision (ICCV). – IEEE, 2015. – P. 1440-1448.
3. Liu, W. SSD: Singe Shot MultiBox Detector / W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg // Computer Vision – ECCV 2016. – Springer, 2016. – P. 21-37.
4. Redmon, J. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – IEEE, 2016. – P. 779-788
5. Everingham, M. The PASCAL Visual Object Classes (VOC) Challenge / M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman // International Journal of Computer Vision, Vol. 88, N 2, 2010. – P. 303-338.