

Отчёт об исследовании статьи PixelLink

Лаборатория Гибридных Ителлектуальных Систем МФТИ

25 апреля 2019

Постановка задачи

Одной из актуальных задач глубинного обучения является распознавание сгенерированного на сценах текста. Данную проблему можно решать в два этапа: сначала находить участки (скошенные прямоугольники) с текстом, а потом уже на выделенных областях запускать "читающие" алгоритмы. Цель нашего исследования – повторить результаты статьи PixelLink, авторы которой решают задачу детектирования текста (т.е. первую подзадачу) с помощью сегментационных масок, которые выдает сеть. Маски строятся с помощью связок (links), объединяющих объекты одного класса. Затем для каждой маски находится описывающий её прямоугольник, который выдаётся как результат работы алгоритма. Модель работает на произвольных картинках и выдаёт набор скошенных прямоугольников.

Задачей PixelLink является выделение лишь областей с текстом на фотографиях, а для распознавания текста на этих областях возможно использование любой OCR-модели. Изначально был опробован Google Tesseract OCR, однако результаты оказались весьма удовлетворительными и далёкими от обозначенных в статье цифр. Чтобы понять, причина в Tesseract, недообученности PixelLink или же модель в принципе не способна работать с точностью, обозначенной в статье, было принято решение попробовать использовать другую, более подходящую для распознавания текста на сценах OCR-модель (см. далее).

Описание программной части

За основу реализации был взят код, предоставленный авторами статьи и находящимся в открытом доступе по адресу https://github.com/ZJULearning/pixel_link. Код использовался почти без изменений, за исключением файла `test_pixel_link_on_any_image.py`. Этот скрипт запускает модель на переданном в параметрах датасете. В исходный код был добавлен функционал сохранения результатов. Веса модели были взяты также из основного репозитория PixelLink. Они были предобучены на датасете ICDAR2015 Challenge 4.1 [3] и использовались в таком виде для дальнейших экспериментов.

В качестве OCR-моделей использовались Google Tesseract OCR и Visual Attention based OCR by Qi Guo and Yuntian Deng, взятый из открытого репозитория <https://github.com/da03/Attention-OCR>.

Технические характеристики машины, на которой проводились эксперименты, приведены на скриншоте ниже.

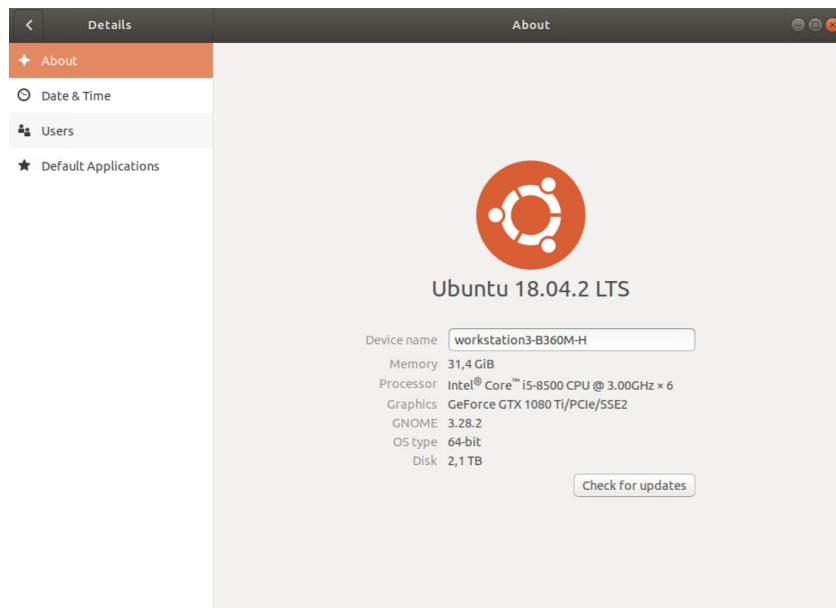


Рис. 1: Технические характеристики сервера

Описание данных

Тестовые данные были взяты из трех датасетов различной степени сложности:

1. ICDAR2015 Challenge 4.1 [3].

Набор данных содержит 500 тестовых изображений, снятых на Google Glass и содержащих текст, представленный в различных ракурсах. Для каждого изображения имеется разметка, состоящая из набора скошенных прямоугольников. Они задаются координатами углов, перечисленных по часовой стрелке. Также представляется верный текст для каждого из прямоугольников. В случае, если текст неразборчив, слишком мал или состоит из менее, чем 3 символов, то он отмечается символами ###, и соответствующий прямоугольник считается неважным и не участвует в оценке.

Пример изображения из данного набора можно увидеть на рисунке 3 (в левом верхнем углу).

2. ICDAR2013 Focused Scene Text [4].

Данный датасет содержит 223 тестовых изображения. На каждом из них предмет с текстом является главным объектом фотографии. Разметка для данного набора представлена в виде множества прямоугольников, ориентированных по осям картинки и задающихся четыремя координатами. Для приведения тестовых данных к стандартному виду скошенных прямоугольников был реализован скрипт convert_4_points_to_8.py, получающий на вход путь до датасета ICDAR2013 и преобразующий его к виду ICDAR2015. Пример изображения показан на рисунке 2.

3. Синтетические данные

Для генерации данных была использована библиотека SynthText [2] и набор изображений, карт глубин и текстов, распространяемых с ней. Полный набор данных содержит 800 тысяч изображений. Из них случайным образом было выбрано 500 картинок и на каждую были встроены несколько текстовых изречений, выбранных из 8 миллионов доступных. Для полученных файлов была также сохранена разметка со скошенными прямоугольниками. Пример изображения можно видеть на рисунке 3 (правый верхний угол).

Методология оценивания

Тестовые изображения подавались на вход предобученной модели, которая выдавала на выходе для каждой картинки набор скошенных прямоугольников — предполагаемое местоположение слов. Полученные прямоугольники сравнивались с тестовыми следующим образом: Для каждого элемента из декартового произведения $\text{predictedboxes} \times \text{groundtruthboxes}$, то есть для всех различных пар прямоугольников, один из которых предсказан, а второй является тестовым, вычислялось значение intersection-over-union (IoU). Если IoU принимал значение ≥ 0.5 , то два данных прямоугольника считались совпадающими. Заметим, что несколько предсказанных прямоугольников могли одновременно совпадать с одним из тестовых, равно как и несколько тестовых с одним из предсказанных. Обе ситуации не запрещались, и все участвовавшие прямоугольники отмечались.

Отдельно остановимся на процедуре подсчёта IoU. Так как, в отличие от осевых прямоугольников, совместное расположение произвольно повёрнутых прямоугольников может быть разнообразным, то использовался не самый эффективный, но точный метод подсчёта площадей пересечения и объединения. Сначала для каждой пары прямоугольников находился описывающий их осевой прямоугольник. Затем для каждого пикселя внутри описывающего прямоугольника происходила проверка, лежит ли он внутри обоих прямоугольников, лежит ли только в одном или вообще не лежит. После проверки, соответствующим образом обновлялось значение площадей. После нахождения соответствий вычислялось количество совпадающих прямоугольников из тестовой разметки (True Positive, TP), количество несовпадающих прямоугольников из тестовой разметки (False Negative, FN) и количество несовпадающих предсказанных прямоугольников (False Positive, FP).

Эти величины использовались для посчёта нескольких метрик.

1. $\text{Precision} = \frac{TP}{TP+FP}$ Точность показывает, насколько предсказанные прямоугольники действительно являются текстом.
2. $\text{Recall} = \frac{TP}{TP+FN}$ Полнота указывает долю правильно отмеченных прямоугольников из тестовой выборки, то есть насколько много объектов детектор находит.
3. $\text{F1 score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$ F1 мера является сбалансированной относительно точности и полноты метрикой, показывающей насколько, в целом, детектор справляется с своей задачей.

Все метрики считались в макро- и микро-режиме. Микро-режим подразумевает подсчёт трех метрик для каждой из картинок, а затем их усреднение по всем изображениям. В макро-режиме, в свою очередь, вычисляются значения ТР, FN и FP для всех прямоугольников всех изображений, а уже затем эти значения используются для вычисления метрик. Данные метрики оценены для всех трех датасетов и представлены в следующем разделе. Исходный код, производящий вычисление этих метрики, представлен в файле `evaluate_detections.py`. Помимо этого для датасета ICDAR 2015 было косвенно оценено качество получаемых прямоугольников. Для этого, все предсказанные прямоугольники были вырезаны из исходного изображения, и на них был запущен алгоритм распознавания текста Tesseract [5]. Затем между полученным результатом и правильным текстом было посчитано расстояние Левенштейна. Для полноты картины также представляется расстояние Левенштейна, нормализованное на максимальную длину двух слов. Результаты представляются как в микро-, так и макро-режиме.

Результаты обучения PixelLink

Основные результаты показаны в таблице 1. Видим, что значения метрик сильно меньше, чем представленные в статье. В таблице 2 представлены расстояния Левенштейна для датасета ICDAR 2015 (PixelLink + Tesseract). И здесь цифры оставляют желать лучшего. На рисунках 2 и 3 показаны визуальные результаты обнаружения текста. Нетрудно увидеть слово, окружённое в слишком большой блок, два слова в одном прямоугольнике, пустой прямоугольник, а то и вовсе необнаруженный текст. Основным предположением о том, почему были получены столь удовлетворительные результаты, является гипотеза о недообученности до нужного качества PixelLink. Чтобы убедиться в этом и исключить влияние Tesseract, было решено попробовать более подходящий для нашего исследования Attention-OCR by Guo and Den.



Рис. 2: Изображение из ICDAR2013

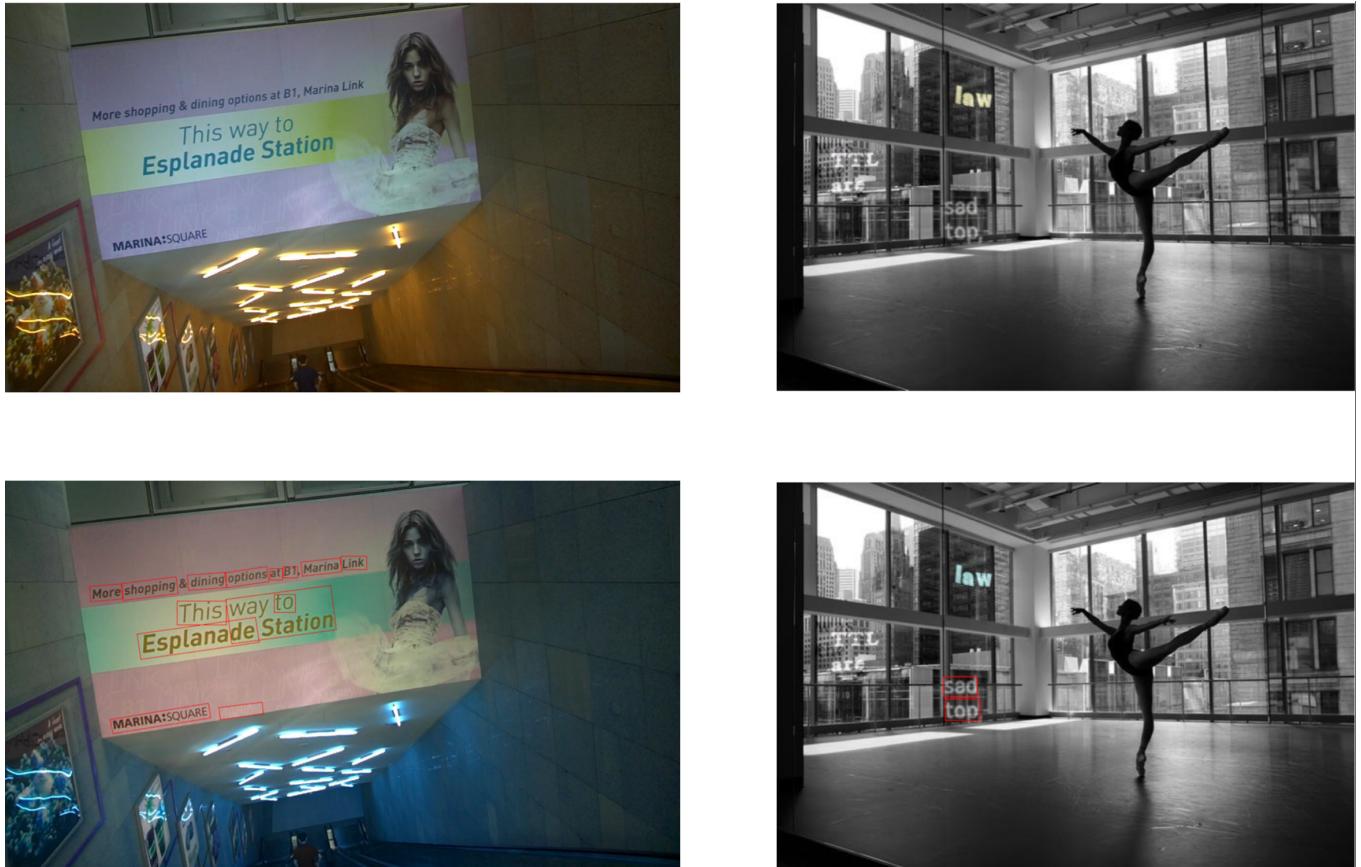


Рис. 3: Результат работы PixelLink на изображении из ICDAR2015 и синтетическом изображении

	Precision		Recall		F1 Score	
	Micro	Macro	Micro	Macro	Micro	Macro
ICDAR 2015	65.18%	63.65%	70.75%	67.00%	65.85%	65.29%
ICDAR 2013	56.33%	59.54%	53.72%	57.26%	52.79%	58.38%
SynthText	64.86%	70.75%	45.62%	46.95%	51.42%	56.45%

Таблица 1: Результаты работы детектора PixelLink

	Micro	Macro
Regular	2.64	2.64
Normalized	0.54	0.55

Таблица 2: Расстояние Левенштейна для найденных текстов на изображениях ICDAR 2015

Оценка Attention-OCR

Обучение модели

Для оценки результатов обучения использовались общедоступные датасеты ICDAR03, IIIT5k и SVT. Каждый из них является просто набором картинок со словами и не нуждается в подробном описании. В качестве метрики использовалось расстояние Левенштейна.



Рис. 4: График обучения модели. Первые пять тысяч итераций. После $\sim 100k$ итераций график редко опускается ниже 99%.

Таблица 3: Результаты работы Attention-OCR

Датасет	icdar03	iiit5k	svt
Точность	95,8%	92,6%	90,7%

```
2019-04-15 09:48:43,146 root INFO    step_time: 0.027230, loss: 0.035732, step perplexity: 1.036378
2019-04-15 09:48:43,148 root INFO    823.981460 out of 860 correct
workstation3@workstation3-B360M-H:~/Desktop/pixel_link_project/Attention-OCR$
```

Рис. 5: Результат работы на датасете ICDAR03

Модель была успешно обучена на датасете SynthText (синтетические изображения, общий объём около 150GB). Уже на половине обучавших изображений точность на тестовых датасетах перестала меняться, поэтому на $\sim 6/10$ объёма SynthText обучение было остановлено. Чекпоинт (три файла, содержащие число 373500) доступен по [ссылке](#).

Результаты обучения можно смело считать успешными хотя бы потому, что авторы другой популярной Attention-OCR модели (находящейся в открытом доступе в github-репозитории tensorflow) после 6 дней обучения смогли получить точность лишь в 83 процента.

Анализ результатов

Посмотрим теперь на те редкие случаи, когда модель ошибалась. Условно их можно разделить на три группы:



Рис. 6: Примеры случаев, когда алгоритм выдавал ошибку.

1. Ошибка из-за специфики шрифта. Например, буква "i" была принята за букву "j" в слове "immortals". Другой пример – слово "louis" было прочитано как "i quis". Данную проблему может решить спеллчекер. Примечательно, что некоторые компании используют для своих эмблем уникальные, ни на что не похожие шрифты. В итоге, кудряво написанный "carlsberg" был прочитан как "crisbery". Вероятно, что тут может помочь только добавление логотипов всех самых известных брендов в обучающую выборку.
2. Ошибка из-за плохого качества изображения. Например, из-за того, что цвет текста и фона совпали, буква "u" в слове "council" была принята за сочетание букв "l" и "i". В слове "beers" у "R" была закрыта правая часть, и она превратилась в "F". А в слове "slavery" эта же буква была настолько смазана, что превратилась в букву "B". Данную проблему также можно решить спеллчекером.
3. Третий, особый случай. Ошибка обусловлена тем, что тестовые изображения содержат начало или конец другого слова. Проблема может быть решена опять-таки спеллчекером, однако, судя по результатам предыдущего отчёта, предобученный на ICDAR2015 PixelLink имеет свойство оставлять слишком много пустого места при выделении областей с текстом, поэтому данному типу ошибок должно быть уделено непосредственное внимание.

PixelLink + Attention-OCR

Имея на руках хорошо обученную OCR-модель, была протестирована связка PixelLink + Attention-OCR и получены следующие цифры (аналогично PixelLink + Tesseract). Результаты записаны в таблицу 4.

	Micro	Macro
Regular		
Normalized		

Таблица 4: Расстояние Левенштейна для найденных текстов на изображениях ICDAR2015. Использована более подходящая OCR-модель

Здесь должен быть вывод.

Заключение

Здесь должно быть заключение.

Список литературы

- [1] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation. 01 2018.
- [2] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, pages 2315–2324. IEEE, 2016.
- [3] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. Icdar 2015 competition on robust reading. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 1156–1160, Aug 2015.
- [4] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras. Icdar 2013 robust reading competition.
- [5] A. Kay. Tesseract: An open-source optical character recognition engine. Linux J., 2007(159):2–, July 2007.

Ссылки

github PixelLink

https://github.com/ZJULearning/pixel_link

github Attention-OCR

<https://github.com/da03/Attention-OCR>

Обученная первая модель

<https://drive.google.com/open?id=1dAIT9zoGNJhN9T3CyTwj0kvs81pHQveB>