Постановка задачи сравнения нейросетевых и непрерывно-морфологических методов детекции текста

Гайдученко Н.Е., Труш Н.А, **Торлак А.В**, Миронова Л.Р., Акимов К.М., Гончар Д.А.

December 9, 2018

Постановка задачи:

В рамках данной статьи были поставлены следующие задачи:

- Рассмотреть наиболее успешные методы обнаружения текста.
- Выделить главные особенности алгоритмов при работе с изображениями плохого качества, а так же фотографий, сделанных под разными углами.
- Реализовать предложенный алгоритм обнаружения текста.
- Провести ряд экспериментов на разных тестовых данных.
- Оценить ошибку каждого из использованных методов.
- Сравнить точность работы всех алгоритмов и выделить методы, реализующую наибольшую точность

Описание базового алгоритма:

1.Detecting Text in Fine-scale Proposals

CTPN представляет собой сверточную сеть, которая принимает на вход изображения любого размера. Текст обнаруживается путем плотного скольжения окошка маленького размера и выводит прямоугольники, которые обводят найденный моделью текст. Детектор скользит через карты conv5, где каждый шаг измеряется 16 пикселями. Предсказанные координаты образует секции, содержащие текстовые предложения. Вертикальные координаты каждой секции определеяются следющим образом:

$$v_c = (c_y - c_y^a)/h_a, \qquad v_h = log_(h/h^a)$$
$$v_c^* = (c_y^* - c_y^a)/h_a, \qquad v_h = log_(h^*/h^a)$$

v - предсказанные координаты

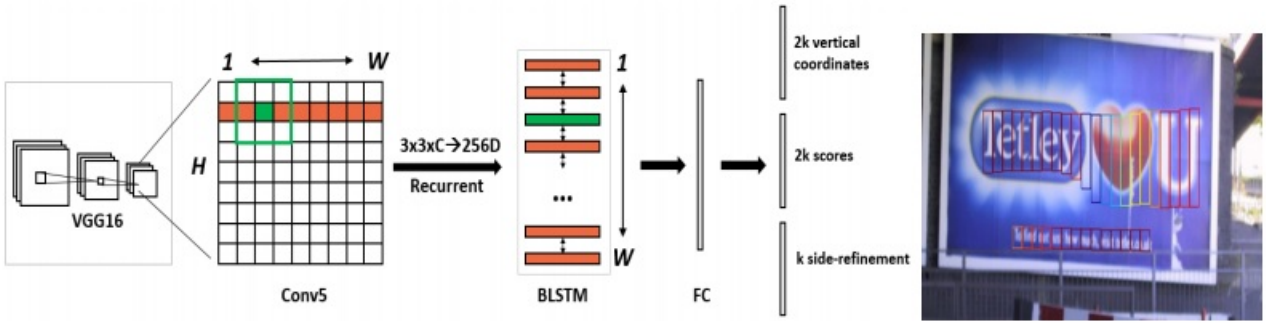
v* - ground-thruth координаты

c_y^a - центр по оси y

h_a - ширина секции

c_y, h - предсказанные координаты

Таким образом, получаем каждое текстовое предложение в определенной рамке размером hx16 как на данном изображении:



2.**Recurrent Connectionist Text Proposals**

Чтобы повысить точность локализации текста, детектируемая строка делится на более мелкие текстовые предложения, каждые из которых обрабатываются отдельно. Для этого , RNN сеть, использующая conv5-карты , принимает на вход сверточную функцию каждого окна как последовательность входов и обновляет его внутреннее состояние в скрытом слое Ht.

$H_t = \phi(H_t - 1, X_t) \qquad t = 1, 2, ..., W$

H_t - рекуррентное внутреннее состояние, которое вычисляется через текущее состояние Xt и предыдущее $H_t - 1$.

X_t - входящий conv5 признак из t скользящего окна(3x3).

W- ширина окна conv5

3.**Side-refinement**

На данном этапе анализируются все детектируемые текстовые секции и принимается каждая текстовая секция у которой соотношение текст/не текст > 0.7.

$o = (x_{side} - c_x^a)/w^a, \qquad o^* = (x_{side}^* - c_x^a)/w^a$

x_{side} - предсказанная x координата до ближайщей горизонтальной стороны(левой или правой) c_x^a - центр координаты w_a - ширина , которая зафиксирована.

Описание Эксперимента:

В качестве набора данных для эксперимента были выбраны следующие датасеты: Multi-lingual scene text, Street View Text. SVT датасет использует

изображения, полученные из сервиса Google Street View. Содержит в себе список слов, которые могут быть детектированы на предложенном изображении. MLT предлагает изображения с текстом на 9 языках и 37 различными шрифтами. Был запущен вышеописанный алгоритм CTPN, для сравнения также был использован EAST.

Анализ ошибки:

Предложенный CTPN метод имеет три основных выходных параметра. Каждый из которых независимо выдает свою оценку текст/не текст. Для улучшения точности используются три функции потерь: L_s, L_o, L_v. Используя три данные функции мы получаем общую функцию потерь, которую необходимо минимизировать:

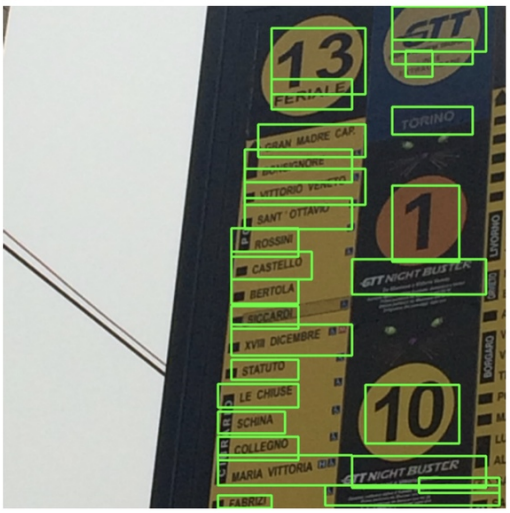
$$L(s_i, v_j, o_k) = \frac{1}{N_s} \sum_{i=1} L_s(s_i, s_i*) + \frac{\lambda_1}{N_v} \sum_{j=1} L_v(v_j, v_j^*) + \frac{\lambda_2}{N_o} \sum_{k=1} L_o(o_k, o_k^*)$$

Коэффициент L_s отвечает за функцию потерь классификации. L_v и L_o - потери регрессий. s_i - прогнозируемая вероятность того, что анализируемый объект i является текстом. j - индекс объекта в наборе всех возможных объектов для координат y регрессии. v_j и v_j^* являются прогнозом и эталонными координатами соответственно для j-го объекта. o_k и o_k^* - предсказанные смещения по оси абсцисс, соответствующие объекту j.

Выбор Модели :

В сравнении с моделью EAST, CTPN оказалась гораздо более точной. Но CTPN требует больше вычислительных ресурсов и работает значительно дольше , нежели EAST. Более высокая точность метода достигается тем, что каждый сегмент текста анализируется отдельно.Затем происходит оценка текст/не текст. В среднем время, затрачиваемое на одно изображение для CTPN - 31s, а EAST в свою очередь детектирует текст за 14s.

EAST **CTPN**



Заключение:

В рамках данной статьи были получены следующие результаты: Рассмотрена область машинного обучения по извлечению признаков. Реализованы несколько базовых алгоритмов для обнаружения текста на изображении, проведен их сравнительный анализ и поставлен ряд экспериментов. Было произведено сравнение работы нескольких моделей : непрерывно морфологических, а также нейросетевых. В ходе эксперимента была замечена зависимость успешности каждого метода от используемого набора данных. Дальнейшие действия направлены на совершенствование архитектуры модели и исследование ее поведения на других данных.