Сравнение 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.

Постановка задачи

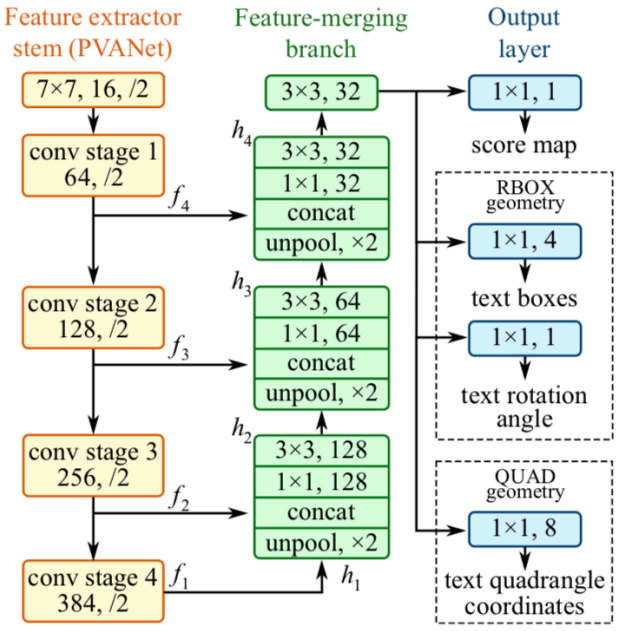
В рамках данной статьи были поставлены следующие задачи:

- 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

Ключевым компонентом предлагаемого алгоритма является модель нейронной сети, которая обучена непосредственному поиску существования текстовых элементов и определению их геометрии на изображениях. Это исключает промежуточные этапы, такие как: поиск кандидата, формирование текстовой области и разбиение слов.

Модель **EAST** может быть представлена в виде трех основных частей, схема приведена на рис. 1



1. Feature extractor stem

База представляет собой сверточную сеть, предварительно обученную на наборе данных ImageNet[1]. Четыре уровня feature map, обозначенных как f_i , извлекаются из базы. Их размеры равны 1/32, 1/16, 1/8 и 1/4 от исходного изображения соответсвенно.

2. Feature-merging branch

Далее мы постепенно объединяем их, пользуясь формулой

$$g_i = \begin{cases} \text{unpool}(h_i) & \text{if} \quad i \leq 3 \\ \text{conv}_{3\times3}(h_i) & \text{if} \quad i = 4 \end{cases} \tag{1}$$

$$h_i = \begin{cases} f_i & \text{if} \quad i = 1 \\ \text{conv}_{3\times3}(\text{conv}_{1\times1}([g_{i-1}; f_i])) & \text{otherwise} \end{cases} \tag{2}$$

где g_i - основание, а h_i - влитая feature map (оператор [x;x] является конкатенацией)

На каждом этапе слияния - feature map с предыдущего этапа проходит через unpool слой, увеличивающий ее размер в 2 раза. Далее она конкатенируется с текущей feature map. Затем, слой *conv1×1*[2] сокращает число каналов и объем вычислений, за которым следует слой *conv3×3*, который объединяет информацию для окончательного получения выходных данных этой стадии слияния. После последней стадии слияния слой *conv3×3* создает окончательную feature map ветви слияния и передает ее на выходной слой.

3. Output layer

Количество выходных каналов для каждой свертки показано на рис. 2

Geometry	channels	description
AABB	4	$\mathbf{G} = \mathbf{R} = \{d_i \vert i \in \{1, 2, 3, 4\}\}$
RBOX	5	$\mathbf{G} = \{\mathbf{R}, \theta\}$
QUAD	8	$\mathbf{G} = \mathbf{Q} = \{(\Delta x_i, \Delta y_i) \vert i \in \{1, 2, 3, 4\}\}$

Мы поддерживаем небольшое количество каналов для сверток в ветви, что добавляет небольшую часть накладных вычислительных расходов на стержень и делает сеть эффективной. Выходной слой содержит несколько операций *conv1×1* для проецирования 32 каналов feature map в 1 канал score map *Fs* и многоканальную *Fg*. Для RBOX - геометрия представлена 4 каналами выровненной по осям прямоугольной

рамкой (AABB) $\boldsymbol{R}$ и 1 каналом, который представляет угол поворота ϑ.
Для QUAD $\boldsymbol{Q}$ используется 8 чисел для обозначения сдвига координат из четырех угловых вершин $\{p_i \mid i \in \{1,2,3,4\}\}$ четырехугольника к месту расположения пикселя. Поскольку каждое смещение расстояния содержит два числа $(\Delta x_i, \Delta y_i)$ то , то вывод содержит 8 каналов.

Планирование эксперимента

В качестве набора данных для эксперимента были выбраны следующие датасеты:

- Multi-lingual scene text [3]
- Street View Text [4]

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 EAST, для сравнения также был использован CTPN.

Анализ ошибки

Ошибка эксперимента может быть представлена следующей формулой:

$$L = L_{\mathrm{s}} + \lambda_{\mathrm{g}} L_{\mathrm{g}}$$

где Ls - представляет ошибку метрики, Lg - ошибку геометрии. λg - весовой коэффициент между ними, в наших экспериментах мы установили значение $\lambda g = 1$. Сбалансированная потеря кросс-энтропия представлена:

$$\begin{aligned} L_{\mathrm{s}} &= \text{balanced-xent}(\hat{\mathbf{Y}}, \mathbf{Y}^*) \\ &= -\beta \mathbf{Y}^* \log \hat{\mathbf{Y}} - (1-\beta)(1-\mathbf{Y}^*) \log(1-\hat{\mathbf{Y}}) \end{aligned}$$

$\hat{\mathbf{Y}}$ - прогноз результатов, **Y*** - эталон, β - коэффициент балансировки.

$$L_{\mathrm{g}} = L_{\mathrm{AABB}} + \lambda_\theta L_\theta .$$

$\boldsymbol{\lambda}_\vartheta = 10, \mathbf{L}_\vartheta$ - ошибка угла поворота, за $\mathbf{L}_{AABB}$ приняты потери IoU, так как они инвариантны относительно объектов различных масштабов.

$$L_{\mathrm{AABB}} = -\log \mathrm{IoU}(\hat{\mathbf{R}}, \mathbf{R}^*) = -\log \frac{|\hat{\mathbf{R}} \cap \mathbf{R}^*|}{|\hat{\mathbf{R}} \cup \mathbf{R}^*|}$$

$\hat{\mathbf{R}}$ - прогнозируемая геометрия AABB, $\mathbf{R}^*$ - представляет эталон.

Выбор модели

В сравнении с моделью CTPN, EAST оказалась менее точной. Но CTPN требует большее количество вычислительных ресурсов и работает значительно дольше, нежели EAST. Наглядная разница представлена на рис. 3

EAST **CTPN**

Более высокая точность **CTPN** была достигнута тем, что каждый сегмент текста анализируется отдельно. Затем происходит оценка текст/не текст. В среднем время, затрачиваемое на одно изображение **CTPN** *31s*, когда для **EAST** среднее время составило *14s.*

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.

---

[1] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei- Fei. Imagenet: A large-scale hierarchical image database. In Proc. of CVPR, 2009.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv: 1512.03385, 2015.

[3] MLT dataset http://rrc.cvc.uab.es/?ch=8&com=downloads

[4] SVT dataset http://vision.ucsd.edu/~kai/svt/