

О вариантах решения задачи распознавания табличной структуры по изображению в условиях отсутствия априорной информации

Н.О. Бешапошников ^{1,A}, А.Г. Леонов ^{2,A,B,C}, М.А. Матюшин ^{3,A}

^A Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук

^B Федеральное государственное образовательное учреждение высшего образования Московский государственный университет имени М.В.Ломоносова

^C Московский педагогический госуниверситет

¹ ORCID: 0000-0002-7616-3143, nbesshaposnikov@vip.niisi.ru

² ORCID: 0000-0001-9622-1526, dr.l@vip.niisi.ru

³ ORCID: 0000-0003-1775-6894, itsaprank@yandex.ru

Аннотация

В данной статье рассматривается задача распознавания табличной структуры по изображению. Постановка проблемы следующая: имеется изображение, на котором запечатлено заведомо неизвестное количество определенных объектов, находящихся в упорядоченной плоской структуре - таблице. На изображение накладываются разумные ограничения на перспективные искажения и повороты. Целью является распознавание заложенной в изображении табличной структуры, то есть расстановка заведомо расположенных в одной плоскости объектов в ячейки таблицы по изображению, или по-другому, табулирование. Данная статья посвящена поиску решения задачи табулирования в условиях отсутствия какой-либо информации о табличной структуре, заложенной в изображении, помимо самого изображения.

Ключевые слова: табулирование, машинное обучение, глубокое обучение, нейронные сети, распознавание изображений, таблица, упорядочивание объектов.

1. Введение

Известно [1], как с помощью технологий глубокого обучения можно эффективно табулировать объекты при условии, что размеры сетки таблицы заранее известны, то есть, когда при восстановлении целочисленных координат объектов известны целочисленная ширина и высота таблицы в клеточках. В условиях отсутствия подобной априорной информации задача несколько усложняется, потому что встает дополнительный вопрос об определении размеров таблицы. Модель, отвечающую за определение размера таблицы, представленной на изображении, будем в дальнейшем называть сайзером.

В данной статье рассказано о попытках решения задачи определения размера с помощью существующих нейронных сетей, описанных в [1], о смене подхода к задаче определения размера, изменении методики обучения, а также описаны примененные модели сайзера в хронологическом порядке их конструирования, вместе с результатами их работы на тестовом наборе данных.

2. Постановка задачи

В последнее время получила широкое распространение задача распознавания таблиц и табличных структур. В наиболее общей постановке модель получает на вход изображение, содержащее таблицу. Как правило, таблица на изображении напечатана или нарисована на каком-либо плоском графическом носителе, например, бумаге или меловой доске. В качестве выхода модели ожидается описание табличной структуры, то есть объекты (чаще всего текстовая информация), расположенные в ячейках таблицы, и ассоциированные с ними индексы этих ячеек. Часто изображение является сканом или даже сверстанным текстом. Тем не менее, общая постановка задачи не накладывает таких ограничений, заставляя считать возможными также перспективные искажения табличной структуры.

Известным методом решения является разделение алгоритма на две части: детектирование таблицы и распознавание заложенной в ней табличной структуры. На первом этапе алгоритма в этом случае обнаруживаются границы отдельных элементов, которые затем на втором этапе используются для восстановления индексов ячеек, в которых они находятся. В ходе наших исследований второй этап также был разбит на две последовательные ступени: определение относительных табличных координат элементов и определение клеточного размера таблицы. Как нетрудно понять, решение данных двух подзадач составляет полное решение задачи распознавания табличной структуры.

Описание решения задачи определения относительных табличных координат элементов было достаточно подробно изложено в [1]. Таким образом, в данной статье рассматриваются варианты решения второй подзадачи - задачи определения клеточного размера таблицы. С этой целью мы применяем наработанную технику машинного обучения с тем чтобы найти оптимальную архитектуру модели сайзера.

3. Обзор работ в данной области

Последнее время описанная задача неоднократно обсуждалась в литературе, в частности, касательно применения нейросетевых технологий и машинного обучения применительно к решению данной проблемы ([2], [3], [4], [5]). В большинстве своем данные работы рассматривают задачу распознавания табличной структуры в суженной постановке, в которой под изображением понимается изображение отсканированного документа. В качестве решения предлагается использовать глубокие нейронные сети, состоящие, как правило, из нескольких последовательно соединенных сетей известных архитектур ([2]), сводящие задачу распознавания табличной структуры к задаче детектирования объектов. Множество различных алгоритмов, как правило, основанных на технологии сверточных нейронных сетей ([3]), показали весьма высокие результаты на известных тестовых выборках. Отдельные подходы отличаются весьма оригинальными идеями касательно приведения черно-белых изображений документов к традиционному трех-четырёх канальному с тем чтобы использовать известные архитектуры сетей детекции объектов ([4]), таким образом используя технологии transfer learning [6], решая исключительно проблему детекции таблицы. Решения задачи детекции также описано в парадигме обучения без учителя [5]. Также в [7] был предложен ряд классических моделей для использования в качестве базового уровня точности, а также достаточно обширный набор данных с изображениями таблиц, состоящий из реальных документов, набранных в Word и LaTeX.

Тем не менее, в наших экспериментах подобные глубокие архитектуры показывали весьма плохую сходимость при использовании вне рамок узкой постановки задачи детектирования таблиц, в том числе, в задачах распознавания табличной структуры вне контекста документов. В связи с этим, было принято решение разбить задачу на более мелкие подзадачи и решать их по отдельности, применяя уже к ним методы машинного и глубинного обучения.

4. Методы решения

4.1. Задача определения размера как задача регрессии

Среди прочего, в [1] была предложена эффективная нейросетевая модель, табулирующая объекты при априорно известных размерных параметрах таблицы. Однако, данная оговорка является существенным ограничением возможностей применения построенной модели. Чтобы расширить область применимости описанного решения, было построено его естественное обобщение, дополняющее условие задачи табулирования неизвестными параметрами высоты и ширины таблицы, которые предлагается искать, как непрерывные неизвестные, то есть регрессионными методами.

Один из естественных подходов к решению данной задачи - попытаться построить одну единственную модель, которая бы успешно справлялась со всей задачей целиком. В условиях поставленной задачи была опробована нейронная сеть, предсказывающая и размеры таблицы, и относительные координаты объектов за один проход. Архитектура построенной комбинированной модели представлена на рис. 1. Здесь и далее наименования слоев следующие:

MaxPool
Convolutional transposed
Convolutional
Linear
Softmax

Субдискретизирующий слой с функцией максимума [9]
Обратный сверточный слой [8]
Сверточный слой [8]
Полносвязный слой [10]
Слой softmax [11]

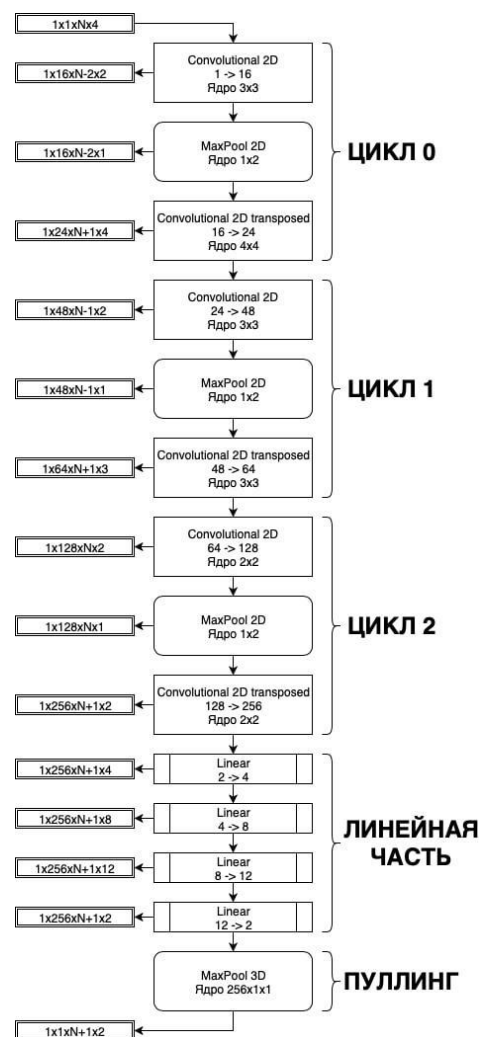


Рис. 1: Архитектура комбинированной модели

На рис. 1 слева от слоев приведены размеры выходных тензоров. Данная сеть получает на вход тензор размерности $1 \times 1 \times N \times 4$, и возвращает тензор размерности $1 \times 1 \times N + 1 \times 2$, здесь N - количество табулируемых объектов. Первые N строк выходного тензора трактуются как относительные координаты табулируемых объектов в таблице, последняя строка выходного тензора трактуется как ширина и высота таблицы.

Легко видеть, что комбинированная сеть является модификацией нейронной сети TabCNNf, описанной в [1]. Разница между этими сетями состоит в наличии дополнительного цикла эмбединга ЦИКЛ 0, а также в большем числе каналов в комбинированной модели. Поскольку данная сеть вычисляет все необходимые данные за один проход, становится возможным её применение на слабощных вычислительных машинах.

4.2. Переформулировка задачи

Выше было описано решение задачи определения размера в случае, когда задача ставится как задача регрессии. Однако, учитывая ограничения на целевой размер таблиц, кажется целесообразным переформулировать задачу. Далее будем решать задачу определения размера как задачу классификации. Будем считать, что на каждую из сторон таблицы наложены ограничения по величине, не более 12 единиц. Таким образом, задача состоит в классификации каждого изображения на принадлежность к одному из $12 \times 12 = 124$ классов. Модели сайзера получают на вход не сами изображения, а нормированные в соответствии с алгоритмом, описанным выше, координаты объектов, найденных на изображениях.

Тем не менее, классификация на такое большое количество классов существенно усложняет задачу, поэтому для упрощения также разобьем задачу на две параллельные задачи классификации: предсказание ширины и предсказание высоты таблицы. Таким образом, под сайзером понимается одна или несколько моделей, которые решают одновременно две задачи классификации по одному изображению: классификацию ширины и классификацию высоты изображенной таблицы.

4.3. Рекуррентный сайзер

В наших экспериментах использовалась модель из двух ячеек LSTM [12] и регуляризацией в виде dropout [13] с вероятностью 0.5 между ними, с последующим полносвязным блоком преобразования к необходимому размеру. Точная архитектура рекуррентного сайзера проиллюстрирована на рис. 2.

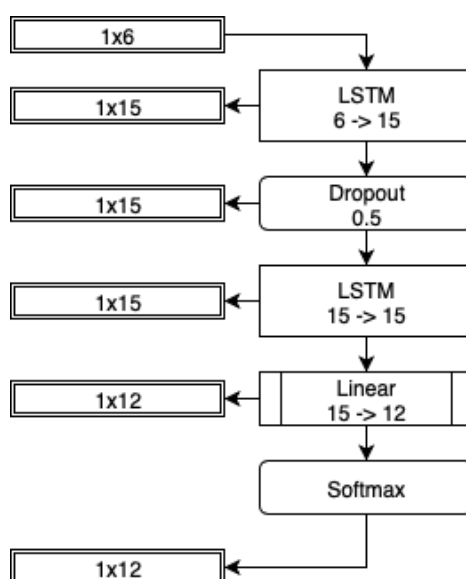


Рис. 2: Архитектура рекуррентного сайзера

Отметим, что за счет использования рекуррентной модели удалось избавиться от неопределенности в размерности входного тензора, так как теперь входной тензор имеет фиксированную размерность 1×6 . От изображения к изображению меняется лишь количество проходов сети (которое равно N - количеству задетектированных объектов).

Скрытое состояние ячеек обнуляется при вычислении очередного изображения, и меняется в ходе вычисления на N строках тензора $N \times 6$.

Для модели рекуррентного сайзера был получен результат $f1\ score = 0.84$ на тестовой выборке, карты $precision/recall$ на тестовой выборке приведены на рис. 3.

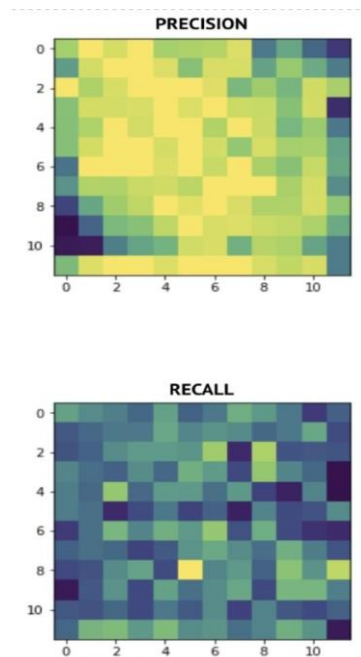


Рис. 3: Точность и полнота рекуррентного сайзера по классам ширины и высоты

Заметим также, что для ширины и высоты были обучены две независимые модели, работающие на одних и тех же исходных данных. Несмотря на впечатляющую по сравнению с описанными в разделе о регрессии моделями, точность рекуррентного сайзера, качество его работы напрямую зависит от количества найденных объектов, что означает ухудшение результатов при работе с малыми и разреженными таблицами.

4.4. “Гистограммный трюк”

С целью фиксации размерности входного тензора также можно применить преобразование входных данных, известное как “гистограммный трюк” [14], [15]. Его суть состоит в следующем: по каждому из 6 признаков строится гистограмма распределения по объектам, затем полученные гистограммы используются в качестве входных данных для модели. Гистограммный трюк решает проблему неопределенности размера, поскольку признаки, как описано ранее, предварительно нормированы в диапазон $[-0.5, 0.5]$ для координат углов объектов и в диапазон $[0, 1]$ для площадей неперекрытых проекций, следовательно, можно строить гистограммы на фиксированных интервалах и с фиксированным шагом, что выливается в конечном итоге в фиксированный размер гистограммы. Здесь и далее количество столбцов гистограмм было принято равным 300. Эксперименты с увеличением данного числа не показали прироста в качестве распознавания, однако в случае работы с экстремально большими таблицами может потребоваться изменение данного значения.

4.5. Сайзер без учителя

Теперь, когда исходные данные имеют заранее определенный размер 300×6 , появляется возможность использования более простых и классических подходов к решению данной задачи. В частности, в рамках решения задачи определения размера были опробованы методы машинного обучения без учителя. В подходах, описанных ниже, входные данные были разделены по координатам на 2 набора, каждый размерности 300×3 . Первый подход заключается в построении по признакам 24 моделей смеси нормальных распределений. Модели традиционно обучаются с помощью EM-алгоритма [16]. В рамках решаемой задачи производится обучение 12 моделей на каждое измерение таблицы. Для вычисления параметров таблицы предлагается следующий подход: распределение данных по гистограммам сравнивается с выученными моделями, далее по методу максимального правдоподобия выбирается модель, под распределение которой наилучшим образом подходит данное изображение. На рис. 4 в виде гистограмм проиллюстрированы распределения, выучиваемые моделями смеси нормальных распределений. По оси абсцисс здесь отложена относительная координата центра объекта табулирования, по оси ординат количество объектов, имеющих такой центр.

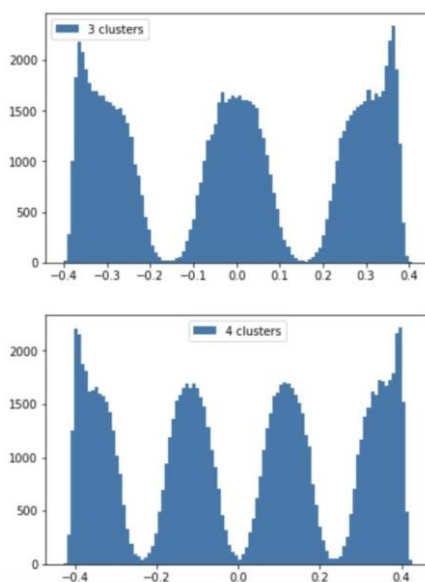


Рис. 4: Распределения координаты центра объекта в зависимости от размера таблицы, по оси абсцисс отложена относительная координата центра объекта, по оси ординат - количество объектов, имеющих такой центр

Несмотря на кажущуюся простоту распределений, при большом значении параметра таблицы распределение становится близко к равномерному, что обуславливает большую ошибку данной модели в том числе и в наших экспериментах.

Также был проверен подход с применением анализа silhouette [17], суть которого заключается в следующем: объекты кластеризуются с переменным числом кластеров алгоритмом KMeans, для каждого числа кластеров вычисляется мера качества кластеризации silhouette, далее предсказанный параметр таблицы считается равным числу кластеров с наилучшим silhouette.

4.6. Полносвязный и сверточный сайзеры

Для решения данной задачи теперь уже классификации также были применены простейшие нейросетевые модели. Первая модель является полносвязной нейронной сетью, её архитектура проиллюстрирована ниже на рис. 5:

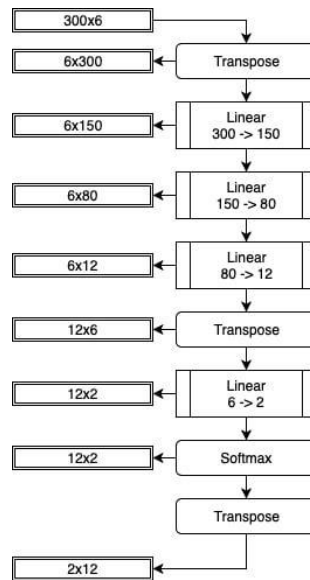


Рис. 5: Архитектура полносвязного сайзера

Вторая модель является сверточной нейронной сетью, её архитектура проиллюстрирована на рис. 6.

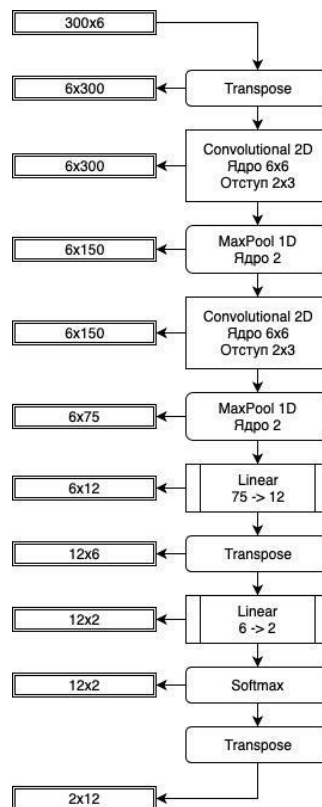


Рис. 6: Архитектура свёрточного сайзера

Выходной тензор в обеих моделях интерпретируется как распределение вероятностей для ширины и высоты по 12 классам.

4.7. Градиентный бустинг, деревянный сайзер

Заключительной моделью в данном исследовании является модель градиентного бустинга [18]. Были построены модели градиентного бустинга на деревьях высоты 2, 3, 5. Качество на валидации монотонно возрастало с увеличением высоты деревьев. Количество деревьев в бустинге равнялось 1000. Существенным отличием в процессе обуче-

ния от предыдущих моделей выступает фиксация тренировочной выборки. Генеративностью тренировочного набора данных пришлось поступить из-за особенностей процесса обучения бустинга. Финальное тестирование, тем не менее, производилось на генеративной выборке. Результаты точности в процессе обучения в зависимости от количества деревьев показаны на рис. 7. Результаты по точности и полноте на тестовой генеративной выборке проиллюстрированы ниже на рис. 8 и рис. 9.

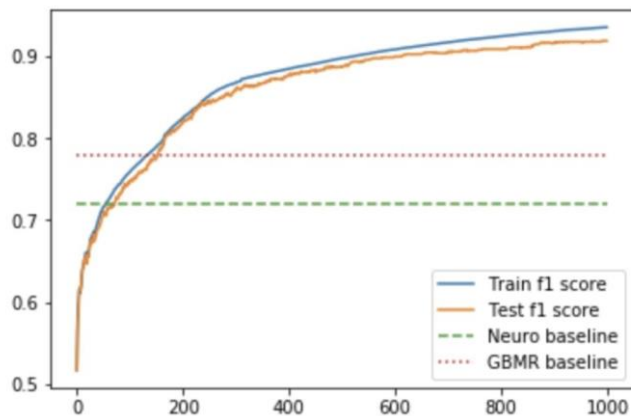


Рис. 7: Обучение градиентного бустинга с 1000 деревьями глубины 5

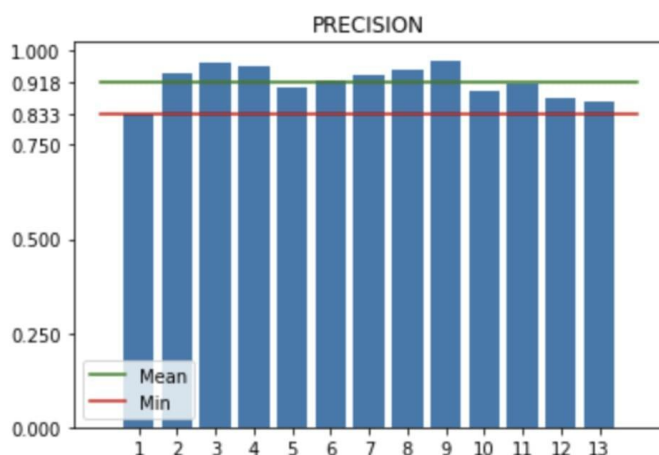


Рис. 8: Точность деревянного сайзера

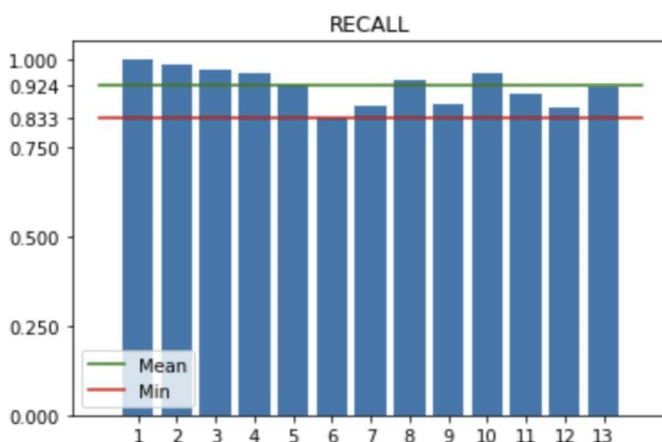


Рис. 9: Полнота деревянного сайзера

5. Экспериментальное тестирование

Исходные данные, поступающие на вход нейросети, также как и выходные данные, подвергаются предварительной нормировке.

Пусть $(x_1, y_1), (x_2, y_2)$ - координаты левого верхнего и правого нижнего углов объекта на изображении, (x_0, y_0) - координаты левого

верхнего угла наименьшей области изображения, в которой находятся все задетектированные объекты, w, h — ширина и высота данной области. Обозначая нормированные координаты через $(\underline{x}_1, \underline{y}_1), (\underline{x}_2, \underline{y}_2)$, получаем искомые формулы нормировки в виде (1):

$$\begin{aligned}\underline{x}_1 &= \frac{x_1 - x_0}{w} - 0.5 \\ \underline{x}_2 &= \frac{x_2 - x_0}{w} - 0.5 \\ \underline{y}_1 &= \frac{y_1 - y_0}{h} - 0.5 \\ \underline{y}_2 &= \frac{y_2 - y_0}{h} - 0.5\end{aligned}\tag{1}$$

Для объекта, находящегося в таблице в клетке с координатами (i, j) , обозначим через $(\underline{i}, \underline{j})$ его нормированные табличные координаты. Тогда получим формулы нормировки в виде (2):

$$\begin{aligned}\underline{i} &= \frac{i}{w_c} - 0.5 \\ \underline{j} &= \frac{j}{h_c} - 0.5\end{aligned}\tag{2}$$

Здесь w_c, h_c - ширина и высота таблицы в клетках, объект исследования настоящей статьи. Поскольку w_c, h_c - выходные значения модели, они также нуждаются в нормировке. Обозначим через $\underline{w}_c, \underline{h}_c$ нормированные значения ширины и высоты. Допустим, обучающий и тестовый наборы данных заведомо содержат лишь таблицы размерами $w_c \times h_c$, где $1 \leq w \leq w_c \leq W, 1 \leq h \leq h_c \leq H$.

Тогда формулы нормировки принимают вид (3):

$$\begin{aligned}\underline{w}_c &= \frac{1}{w_c} \\ \underline{h}_c &= \frac{1}{h_c}\end{aligned}\tag{3}$$

Таким образом, согласно формулам выше, производится нормировка диапазона допустимых детектируемых ширины и высоты на отрезок $[-0.5, 0.5]$.

Помимо уже известных признаков в виде координат углов bounding box'a объекта были добавлены два следующих признака: площадь неперекрытой проекции объекта при проецировании на ось Ox и на ось Oy изображения. Сущность данных признаков состоит в следующем: пусть известны координаты точек объектов $\{(x_1^{(n)}, y_1^{(n)}), (x_2^{(n)}, y_2^{(n)}), \dots\}$, где $1 \leq n \leq N$ - номер задетектированного объекта.

Введем следующие обозначения:

$$\begin{aligned}x^n &= \sup(\{x_1^{(n)}, x_2^{(n)}, \dots\}), \\ x^n &= \inf(\{x_1^{(n)}, x_2^{(n)}, \dots\}), \\ y^n &= \sup(\{y_1^{(n)}, y_2^{(n)}, \dots\}), \\ y^n &= \inf(\{y_1^{(n)}, y_2^{(n)}, \dots\}),\end{aligned}$$

и пусть μ - мера Лебега на \mathbb{R} . Тогда площади неперекрытых проекций объекта n на оси Ox и Oy следует считать по формулам (4), (5):

$$P_x(n) = \mu([x_n, x^n] \setminus \cup_{i:y_i < y_n} [x_i, x^i])\tag{4}$$

$$P_y(n) = \mu ([y_n, y^n] \setminus \cup_{i:x_i < x_n} [y_i, y^i]) \quad (5)$$

Учитывая (1),(2),(3),(4),(5), входной тензор состоит из N строк вида $\underline{x}_1, \underline{y}_1, \underline{x}_2, \underline{y}_2, P_x, P_y$ выходной тензор состоит из N строк вида $\underline{i}, \underline{j}$ и последней строки $\underline{w}_c, \underline{h}_c$. для комбинированной модели в регрессионной постановке, или же из одной строки вида $\underline{w}_c, \underline{h}_c$. для остальных описанных моделей в классификационной постановке.

Во всех экспериментах как обучающий, так и тестирующий наборы данных являлись полностью синтетическими, не были фиксированы и генерировались на лету во время тренировки, модель гарантированно ни разу не видела одно и то же изображение.

Таким образом, в экспериментах используется имеется модуль, способный генерировать изображения табличных объектов с поворотами и перспективными искажениями, с параметрами (шириной и высотой) таблицы в диапазоне от 1 до 12, и создающий из этих изображений тензоры размерности $N \times 6$, где N - количество объектов табулирования, 6 – признаковое пространство, состоящее из нормированных координат верхнего левого и правого нижнего углов, а также нормированных площадей непрерывных проекций на каждую из осей изображения и/или осей таблицы. Пример изображения, использованного для обучения и тестирования моделей, проиллюстрирован на рис. 10. Синими цифрами отмечены истинные координаты ячеек, красными - результат распознавания модели.

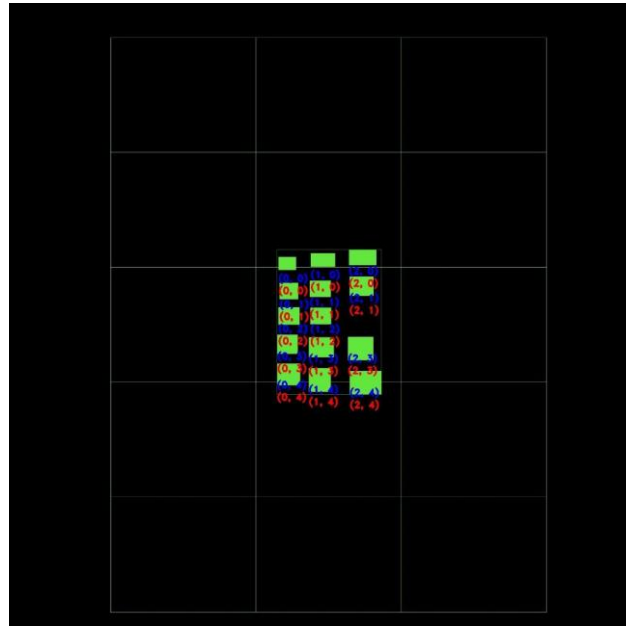


Рис. 10: Пример изображения тестового набора данных

Результаты точности описанных выше моделей измеряются в терминах f1 score, и приведены в таблице ниже:

Метод	F1 score
Модель автокодировщика с расширением тензора	0.1
Кластеризационная модель	0.2
Комбинированная модель	0.6
Полносвязная и сверточная модели	0.71
Рекуррентная модель	0.84
Градиентный бустинг	0.93

6 Заключение

В данной статье был предложен ряд решений задачи определения размера таблицы для дальнейшего табулирования в условиях отсутствия априорной информации.

Полученные результаты показывают, что применение классического градиентного бустинга позволяет в рамках данной задачи получить наилучший результат. Тем не менее, поскольку обучающий набор данных для модели бустинга был фиксирован, в отличие от нейросетевых моделей, она потенциально обладает более низкими обобщающими способностями при дальнейшей эксплуатации в реальных условиях.

Поскольку даже максимальная точность, достигнутая в описанных экспериментах, далека от 1.0, поиск оптимального решения будет продолжен. В частности, следующий цикл экспериментов включает проверку комбинаций описанных моделей, таких как инициализация бустинга при помощи рекуррентной модели, или, например, тренировка рекуррентной модели на признаках, выделенных с помощью бустинга.

Работа выполнена при поддержке Гранта РФФИ 18-07-00901.

Список литературы

1. Бесшапошников Н.О. Леонов А.Г. Матюшин М.А. Вопросы упорядочивания объектов на изображении с использованием нейросетевых и эвристических алгоритмов. // Вестник кибернетики - No 4(32), 2018.
2. Sebastian Schreiber Stefan Agne Ivo Wolf Andreas Dengel Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017.
3. A Table Detection Method for PDF Documents Based on Convolutional Neural Networks / Leipeng Hao, Liangcai Gao, Xiaohan Yi [и др.]. 12th IAPR Workshop on Document Analysis Systems (DAS), 2016.
4. Table Detection Using Deep Learning / Azka Gilani, Shah Rukh Qasim, Imran Malik [и др.]. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017.
5. Fan M. Kim D. S. Table Region Detection on Large-scale PDF Files without Labeled Data. CoRR, abs/1506.08891, 2015.
6. West Jeremy; Ventura Dan; Warnick Sean. A Theoretical Foundation for Inductive Transfer. Spring Research, 2005.
7. Minghao Li Lei Cui Shaohan Huang Furu Wei Ming Zhou Zhoujun Li. TableBank: Table Benchmark for Image-based Table Detection and Recognition. arXiv preprint arXiv:1903.01949, 2019.
8. Alex Krizhevsky Ilya Sutskever Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. // Advances in Neural Information Processing Systems 25, 2012.
9. Yamaguchi Kouichi; Sakamoto Kenji; Akabane Toshio; Fujimoto Yoshiji. A Neural Network for Speaker-Independent Isolated Word Recognition. // First International Conference on Spoken Language Processing (ICSLP 90), 1990.
10. Rosenblatt F. THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. // Psychological review - No 6(65), 1958.
11. F.R. Christopher M. Bishop. Pattern Recognition and Machine Learning. // Springer, 2006.
12. Sepp Hochreiter Jurgen Schmidhuber. Long Short-Term Memory. // Neural computation - No 9(8), 1997.
13. Nitish Srivastava Geoffrey Hinton Alex Krizhevsky Ilya Sutskever Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. // Journal of Machine Learning Research -15, 2014.

14. Chapelle Olivier, Haffner Patrick, Vapnik Vladimir. SVMs for Histogram-Based Image Classification. // IEEE TRANSACTIONS ON NEURAL NETWORKS - No 5(10), 1999.
15. Norbert Obsuszt AnswerMiner. Histogram 202: Tips and Tricks for Better Data Science. <https://www.kdnuggets.com/2018/02/histogram-tips-tricks.html>.
16. UniversityStanford.GaussianMixturesandtheEMalgorithm.
<http://statweb.stanford.edu/tibs/sta306bfiles/mixtures-em.pdf>.
17. J.Rousseeuw Peter. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. // Journal of Computational and Applied Mathematics, 1987.
18. Friedman Jerome H. Greedy Function Approximation: A Gradient Boosting Machine. // The Annals of Statistics - No 5(29), 2001.

Solutions to recognize the table structure by an image in the absence of a priori information

N.O. Besshaposhnikov^{1,A}, A.G. Leonov^{2,A,B,C}, M.A. Matyushin^{3,A}

^A Scientific Research Institute for System Analysis of the Russian Academy of Sciences

^B M.V.Lomonosov Moscow State University

^C Moscow Pedagogical State University

¹ ORCID: 0000-0002-7616-3143, nbesshaposhnikov@vip.niisi.ru

² ORCID: 0000-0001-9622-1526, dr.l@vip.niisi.ru

³ ORCID: 0000-0003-1775-6894, itsaprank@yandex.ru

Abstract

In this paper, we consider the problem of recognizing a table structure through the analysis of the provided picture. The problem statement is the following: we have a photo with an unknown number of particular objects captured, and we know that they are arranged in a flat table structure. It is assumed that the provided picture complies reasonable restrictions concerning perspective distortion and rotation magnitudes. The goal is to recognize the underlying table structure, i.e., to arrange the recognized objects into some table structure that appropriately fits the picture. From now on, we call this procedure the tabulating of the objects. This paper then considers the task of tabulating objects under the conditions of the absence of any antecedent information concerning the table structure, except for the actual picture.

Keywords: tabulating, machine learning, deep learning, neural networks, image recognition, table structure, objects ordering.

References

1. Besshaposhnikov N.O., Leonov A.G., Matyushin M.A. Voprosy uporyadochivaniya objektov na izobrazhenii s ispol-zovaniem neyrosetevykh i ehvristicheskikh algoritmov [On the arranging detected objects using neural networks and heuristic algorithms]. // Proceedings in cybernetics - 4(32), 2018.
2. Sebastian Schreiber Stefan Agne Ivo Wolf Andreas Dengel Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017.
3. A Table Detection Method for PDF Documents Based on Convolutional Neural Networks / Leipeng Hao, Liangcai Gao, Xiaohan Yi [et al.]. 12th IAPR Workshop on Document Analysis Systems (DAS), 2016.
4. Table Detection Using Deep Learning / Azka Gilani, Shah Rukh Qasim, Imran Malik [et al.]. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017.
5. Fan M. Kim D. S. Table Region Detection on Large-scale PDF Files without Labeled Data. CoRR, abs/1506.08891, 2015.
6. West Jeremy; Ventura Dan; Warnick Sean. A Theoretical Foundation for Inductive Transfer. Spring Research, 2005.
7. Minghao Li Lei Cui Shaohan Huang Furu Wei Ming Zhou Zhoujun Li. TableBank: Table Benchmark for Image-based Table Detection and Recognition. arXiv preprint arXiv:1903.01949, 2019.
8. Alex Krizhevsky Ilya Sutskever Geoffrey E. Hinton. ImaeNet Classification with Deep Convolutional Neural Networks. // Advances in Neural Information Processing Systems 25, 2012.

9. Yamaguchi Kouichi; Sakamoto Kenji; Akabane Toshio; Fujimoto Yoshiji. A Neural Network for Speaker-Independent Isolated Word Recognition. // First International Conference on Spoken Language Processing (ICSLP 90), 1990.
10. Rosenblatt F. THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. // Psychological review - No 6(65), 1958.
11. F.R. Christopher M. Bishop. Pattern Recognition and Machine Learning. // Springer, 2006.
12. Sepp Hochreiter Jurgen Schmidhuber. Long Short-Term Memory. // Neural computation - No 9(8), 1997.
13. Nitish Srivastava Geoffrey Hinton Alex Krizhevsky Ilya Sutskever Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. // Journal of Machine Learning Research -15, 2014.
14. Chapelle Olivier, Haffner Patrick, Vapnik Vladimir. SVMs for Histogram-Based Image Classification. // IEEE TRANSACTIONS ON NEURAL NETWORKS - No 5(10), 1999.
15. Norbert Obsuszt AnswerMiner. Histogram 202: Tips and Tricks for Better Data Science. <https://www.kdnuggets.com/2018/02/histogram-tips-tricks.html>.
16. UniversityStanford.GaussianMixturesandtheEMalgorithm. <http://statweb.stanford.edu/tibs/sta306bfiles/mixtures-em.pdf>.
17. J.Rousseeuw Peter. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. // Journal of Computational and Applied Mathematics, 1987.
18. Friedman Jerome H. Greedy Function Approximation: A Gradient Boosting Machine. // The Annals of Statistics - No 5(29), 2001.