Якубов М.С., д.т.н., профессор, Аъзамов Т. Н., PhD, доцент, Султонов Й.У., базовый докторант, Файзиев Р.А., к.ф.-м.н., профессор, Ражабов Н.А., к.ф.-м.н., доцент

РАСПОЗНАВАНИЕ ТЕКСТА В ОБЛОЖКЕ КНИГ С ПОМОЩЬЮ МАРКИРОВКИ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Ташкентский университет информационных технологий, Узбекистан, г.Ташкент, maksadhan@mail.ru, timur\_azamov@mail.ru, yura\_sultonov@mail.ru

Ташкентский государственный экономический университет, Узбекистан, г.Ташкент, zktdiu@yandex.ru

Самаркандский архитектурно-строительный институт, Узбекистан, г.Самарканд, rna18@mail.ru

Yakubov M. S., Doctor of Technical Sciences, Professor, Azamov T. N., PhD, Associate Professor, Sultonov Y. U., basic doctoral student,

Fayziev R. A., Candidate of Physical and Mathematical Sciences, Professor, Rajabov N. A., Candidate of Physical and Mathematical Sciences, Associate Professor

RECOGNIZING TEXT IN BOOK COVERS BY MARKING SEQUENCES

*Tashkent University of Information Technologies, Uzbekistan, Tashkent,* *maksadhan@mail.ru**,* *timur\_azamov@mail.ru**,* *yura\_sultonov@mail.ru*

*Tashkent State University of Economics, Uzbekistan, Tashkent,* *zktdiu@yandex.ru*

*Samarkand Institute of Architecture and Civil Engineering, Uzbekistan, Samarkand,* *rna18@mail.ru*

Аннотация. В этой статье мы представляем систему построения и поиска книг по ими обложками, основанную на методах чтения текста обложек книг. Мы специально разрабатываем нашу модель распознавания текста изображении, используя расширенными параметрами, чтобы ускорить обучение и добиться высочайшей производительности на нескольких тестовых наборах данных. Предлагаемая нами система может значительно сократить количество человеческого труда, необходимого для поиска книг, а также пространство, необходимое для хранения книжной информации.

Abstract. In this article, we present a system for building and searching for books by their covers, based on the methods of reading the text of book covers. We specifically design our image text recognition model using advanced options to accelerate learning and achieve the highest performance on multiple test datasets. Our proposed system can significantly reduce the amount of human labor required to search for books, as well as the space required to store book information.

Ключевые слова. метод чтения текста, распознавания текста изображении, производительности тестовых наборах данных.

Keywords. text reading method, image text recognition, performance of test datasets.

В нашей системе изображения обложек книг идентифицируются на основе распознанного текста, который затем используется в качестве ключевых слов для индексации или поиска в базе данных книг. В процессе запроса наша система полагается только на текстовую информацию, не требуя исходных изображений книги.

На этапе распознавания текста общий подход состоит в том, чтобы сначала сегментируется и распознается каждый символ, а затем вывести прогноз уровня слов на основе языковой модели или комбинации эвристических правил. Применяем гибридный подход, который сочетает CNN с RNN, рассматривая проблему распознавания текста из изображения как задачу последовательной маркировки. CNN внизу изучает особенности изображений, которые затем составляются в виде последовательностей признаков, которые впоследствии передаются в RNN для последовательной маркировки.

На рис. 1 представлена архитектура предложенной модели распознавания текста. Сначала мы генерируем последовательность глубоких функций CNN, F={f1, f2, ..., ft}. Для дальнейшего использования взаимозависимости между функциями разработана модель двунаправленной долгосрочной краткосрочной памяти (B-LSTM) (Hochreiter и Schmidhuber, 1997) накладывается поверх сгенерированных последовательных функций CNN, давая другую последовательность X={x1, x2, ... xt}в качестве конечных результатов. Каждый из xi нормализуется с помощью функции softmax и может интерпретироваться как излучение определенной метки для данного временного шага. С другой стороны, целевое слово Y можно рассматривать как последовательность символов Y={y1, y2, ..., yL} Y = {y1, y2, · · ·, yL}.



Рис. 1. Архитектура предлагаемой модели распознавания. Элемент CNN аналогичен VGG16 (Simonyan and Zisserman, 2014), за исключением того, что шаг слоев объединения настраивается таким образом, что последняя сверточная карта признаков имеет высоту 1. Пакетная нормализация (Ioffe and Szegedy, 2015) добавляется после каждого слой свертки для ускорения сходимости.

**Потеря СТС (Connectionist Temporal Classiﬁcation).**

Последовательности X и Y имеют разную длину t и L соответственно, что затрудняет обучение нашей модели при отсутствии наземной истины для каждого временного шага. Следуя работе Graves и др. (2006), мы принимаем потерю CTC, чтобы позволить RNN обучаться для задачи маркировки последовательностей без точного выравнивания.

Потеря CTC - это отрицательная логарифмическая вероятность вывода целевого слова Y с учетом входной последовательности X:

*CTC(X)=-log P(Y|X)*

Предположим, есть выравнивание a, которое обеспечивает предсказание для каждого шага (пустые или непустые метки) для *X*, и функция отображения B, которая удаляет повторяющиеся метки и пробелы из a., например, *(-, a, a, -, -, b)* будет отображаться как (a, b) (используя - для обозначения пустой метки). Затем P(Y|X) можно вычислить, суммируя все возможные выравнивания a, которые могут быть сопоставлены с Y:

P(Y|X)=aB-1(Y)P(a|X)

и в предположении независимости:

P(a|X)=i=1TP(ai|X) . (1)

Уравнения (1) может быть эффективно вычислено с использованием метода динамического программирования вперед-назад, как описано в Graves (2006). Декодирование (поиск наиболее вероятного Y из выходной последовательности X) может быть выполнено путем поиска луча (Graves and Jaitly, 2014) или просто путем выбора единственного наиболее вероятного прогноза на каждом временном шаге и последующего применения функции отображения B:

*argYmax P(Y|X)B(argamax P(a|X)).*

**Обучение CTC с пошаговым контролем.**

В процессе обучения CTC пустые метки обычно преобладают в выходной последовательности. Непустые метки появляются только как изолированные пики, пример которых изображен на рис. 2. Это следствие алгоритма вперед-назад (Graves и др., 2006). Поскольку мы добавляем пустую метку между каждым символом, существует больше возможных путей, проходящих через пустую метку на заданном временном шаге в прямом-обратном графе CTC. На ранней стадии обучения CTC, когда веса модели инициализируются случайным образом, все пути имеют схожие вероятности. В результате вероятность того, что данный временной шаг будет пустой меткой, намного выше, чем у любого другого, при суммировании всех действительных выравниваний. Кроме того, градиент обратного распространения, вычисленный с использованием прямого-обратного алгоритм будет поощрять появление пустых меток на каждом отдельном временном шаге.



Рис. 2. Изображение слова в наборе обучающих данных (а). Изображение слова с ограничивающими рамками на уровне персонажа (b). Типичная выходная последовательность из модели, обученной CTC. Серая область соответствует вероятности вывода пустой метки на данном временном шаге, а цвета указывают на непустые метки. Каждая непустая метка представляет собой изолированный пик (c). Истина наземного шага (d), генерируемая на основе (b).

Прогнозирование только пробелов приведет к высокой потере CTC, и в результате модель попытается предсказать символы в определенных временных шагах. Однако из-за проблемы с пустой меткой, описанной выше, обычно требуется много итераций, прежде чем непустая метка появится в выходной последовательности во время обучения CTC. Чтобы ускорить процесс обучения, мы вводим пошаговый надзор. Если у нас есть доступ к ограничивающим рамкам уровня символа в каждом изображении слова, то мы сможем определить метку xi на временном шаге i на основе соответствующего ему воспринимающего поля. В наших экспериментах xi присваивается метка zi = yj, если его рецептивное поле перекрывается более чем с половиной площади yj, в противном случае xi назначается пустая метка

Во время обучения целевая функция становится:

$$L(X) = CT C(X) + λLpt(X)$$

$Lpt(X) = \frac{1}{T}\sum\_{i=1}^{T}-logP(z\_{i}|x\_{i})$

где λ - гиперпараметр, предназначенный для уравновешивания двух членов потери. Поскольку наш пошаговый контроль обеспечивает только одно возможное выравнивание, оно не гарантирует точное решение уравнения (1). Таким образом, мы уменьшаем λ на протяжении всего обучения. Вначале λ устанавливается таким образом, чтобы градиенты, вызванные двумя видами потерь, имели одинаковую величину.

Не существует крупномасштабного общедоступного набора данных, который предоставлял бы информацию об ограничивающей рамке на уровне символов, поэтому мы создали наши собственные синтетические данные, аналогичные Jaderberg (2014), за исключением того, что мы продолжали отслеживать местоположение каждого персонажа.

**Обучение CTC со штрафом за декодирование.**

Еще одна проблема обучения ЦОК - разрыв между целевой функцией и критерием оценки. Как указано в Graves and Jaitly (2014), потеря CTC будет пытаться максимизировать логарифмическую вероятность вывода полностью правильной метки Y = {y1, y2, · · ·, yL}.

Другими словами, неверные прогнозы также считаются плохими. Однако не всегда так оценивается производительность модели. Например, для задачи распознавания текста обложки книг люди часто сообщают о расстоянии редактирования между предсказанием и истинностью. Graves и Jaitly (2014) предложили выборочный метод для расчета ожидаемых убытков. Однако задействованный шаг выборки замедлит процесс обучения.

Здесь мы предложили более простое решение для наказания за плохие прогнозы. Представлена обновленная версия CTC:

$W CT C(x) = -logP(Y |X) · L\_{e}(Y, Y\_{D})$,

где *Le* - это функция потерь реального значения (например, расстояние редактирования между двумя строками), а *YD* - декодированный прогноз. В зависимости от выполняемого метода декодирования *YD* может быть разным. Например, можно приблизительно рассчитать *YD* по:

$YD = B(arg max a a P(a|X)) ≈ arg max Y Y P(Y |X)$*.*

W CTC (X) неизбежно будет подчеркивать более длинный Y, поэтому мы предлагаем, чтобы W CTC (X) активизировался после нескольких периодов обучения CTC, когда большинство YD имеют достаточно малое расстояние редактирования по сравнению с Y.

**Коррекция текста.**

Модель распознавания текста нашей системы ожидает обрезанного фрагмента изображения, содержащего одно слово в качестве входных данных. Однако это не всегда может быть выполнено на этапе локализации текста. Созданная ограничивающая рамка иногда бывает неполной, что приводит к распознаванию слов с отсутствующими символами. Более того, пробелы между словами может быть трудно отличить от пробелов между символами. Следовательно, необходим этап постобработки для корректировки выходных данных модели распознавания. В ходе экспериментов мы обнаружили, что стандартная проверка орфографии недостаточно мощна для автоматического исправления, поскольку нам нужно не только исправлять отдельные слова, но и разбивать более крупные строки на слова. Здесь мы обучаем другую RNN для решения этой проблемы, используя модель последовательность-последовательность на уровне символов (Sutskever, Vinyals, and Le, 2014).

Учитывая входное предложение со случайным шумом (удаление, вставка или замена случайно выбранных символов), ожидается, что RNN от последовательности к последовательности выведет правильный. RNN состоит из двух слоев LSTM, каждый со скрытым размером 512. Во время обучения выборочные предложения из набора данных Google объемом один миллиард слов (Chelba 2013) и их зашумленные варианты используются в качестве целей и входных данных. соответственно. На рисунке 3 показано несколько обрезанных образцов изображений, которые демонстрируют эффективность нашей модели коррекции.

Наша модель распознавания обучается сквозным образом, в то время как He и др. обучил часть CNN и часть RNN отдельно. И наша модель распознавания, и модель коррекции обучаются с использованием мини-пакетного стохастического градиентного спуска (SGD) вместе с Adadelta (Zeiler, 2012).



Рис. 3: Примеры распознавания текста, созданные Tesseract (Smith, 2007), наша модель распознавания и модель коррекции. Сверху вниз, слева направо: (a) Tesseract: mum, fUdlCBldUl ItnMnCe, Ahmn, ELIGINEERIM (b) Наше признание: oigital, CalculatorEnnance, Anton, NGINEERIN (c) Наша поправка: digital, Calculator Enhance, Anton, ENGINEERING.

**Эксперименты распознавание текста.**

Чтобы оценить производительность предлагаемого нами метода распознавания текста, мы сообщаем о наших результатах на нескольких тестовых наборах данных: ICDAR 2003 (IC03), SVT и III5K в соответствии со стандартным протоколом оценки, описанным Wang, Babenko, and Belongie (2011). IC03 предоставляет лексиконы по 50 слов на изображение (IC03-50) и все слова тестового набора (IC03-Full). Изображения в SVT и III5K также связаны с лексиконами (SVT-50, III5K-50, III5K-1K).

Мы называем нашу базовую модель, обученную с использованием потерь *CTC*, моделью *Deep Sequential Labeling (DSL)*, в то время как модель, обученная с использованием исправленных функций потерь, *CTC (X) + λLpt*и *W CTC (X) + λLpt*, будут называться *DSL-CL* и *DSL-WL* соответственно.

На рис. 4 показаны потери CTC во время обучения с использованием различных целевых функций. Чтобы выполнить эффективное мини-пакетное обучение, мы группируем выборки данных по длине их Y. Следовательно, можно наблюдать внезапные скачки кривой потерь, что соответствует более длинному Y в этой партии. Как мы видим, добавление потери классификации за тайм-шаг значительно ускорит обучение на ранней стадии. На более позднем этапе, когда λ становится все меньше и меньше, разница между Lpt и без него уменьшается. Однако из-за потери вариации мы все же можем наблюдать определенные преимущества при использовании Lpt.



Рис. 4: Слева: потеря CTC во время обучения с использованием различных целевых функций. Точки - это потеря обучения, а сплошные линии - потеря проверки. Справа: увеличенная версия на более позднем этапе.

В таблице 1 показаны результаты обученной модели распознавания по сравнению с Tesseract (Smith 2007), использованной в Tsai (2011), а также с некоторыми другими современными методами.

**Таблица 1**

**Урезанная точность распознавания слов в нескольких тестовых наборах данных**



Мы видим, что Tesseract плохо работает во всех случаях. Понятно, что искажения в реальных изображениях обложки книг делают распознавание совершенно иной и сложной задачей по сравнению со стандартной системой распознавания текста. Наши модели распознавания превосходят методы с функциями ручной работы (Wang, Babenko, and Belongie, 2011; Lee и др., 2014; Yao и др., 2014) и несколько методов, которые используют глубокие нейронные репрезентации для отдельных символов (Wang и др., 2012; Jaderberg, Vedaldi, and Zisserman, 2014), что указывает на то, что наша модель дает гораздо лучшие результаты для распознавания путем изучения последовательной информации. Наша базовая модель DSL дает лучшие результаты для всех наборов данных, кроме IC03-50, по сравнению с He и др. Мы связываем это с процедурой сквозного обучения и большим объемом синтетических данных во время обучения. Модель Jaderberg (2015) показывает лучшие результаты для IC03-50 и IC03-Full. Однако, поскольку они рассматривают проблему распознавания текста как задачу классификации нескольких классов (количество классов равно количеству слов), их модели трудно обучить и адаптироваться к тексту вне словаря. Ожидается, что наша базовая модель DSL будет иметь производительность, аналогичную Shi и др. 2016, так как у нас схожая архитектура модели и процедура обучения. Мы связываем разницу с различными используемыми синтетическими обучающими данными и настройкой гиперпараметров.

Используя потерю Lpt на временной шаг, наша модель больше не будет выводить изолированные пики непустого прогноза. Более того, из рисунка 5 и таблицы 1 видно, что DSLCL и DSL-WL работают лучше, чем DSL. Мы предполагаем, что прогнозирование пика вредно, поскольку оно запутает модель относительно того, где давать непустой прогноз, а где нет.

DSL-WL связывает или немного превосходит DSL-CL во всех наборах данных, что позволяет предположить, что наша пересмотренная оценка потерь W CT C (X) эффективна.

**Восстановление обложек книги.**

Чтобы оценить эффективность поиска в нашей системе, мы применяем метод оценки, основанный на поиске, аналогичный тому, который использовали Chen и др. (2010) и Tsai и др. (2011). Однако, поскольку у нас есть доступ только к 454 изображениям обложек книг, которые они использовали для запроса, а не ко всей базе данных для поиска, которая содержит 2 300 книг, необходимо создать нашу собственную коллекцию. Основываясь на темах этих 454 книг, мы просмотрели и отобрали более 9000 книг, чтобы создать нашу базу данных вместе с этими 454 тестовыми изображениями позвоночника. Для каждой книги в нашей базе данных мы просканировали ее название и метаданные, такие как автор и издатель. Затем эта информация индексируется для последующих задач поиска. Мы ожидаем, что созданный нами набор данных является надмножеством их, а это означает, что более высокая точность и отзывчивость наших результатов могут указывать на превосходную производительность.

Наша система полагается только на распознавание текста. В результате нам нужна только текстовая информация каждой книги, и нам не нужно хранить и запрашивать изображения обложек, что значительно сокращает объем данных, которые нам нужно хранить и управлять. Более того, для библиотек с ограниченными ресурсами нельзя всегда предполагать, что изображения книг уже доступны.

Для каждого изображения обложек книги обнаруживается, распознается и исправляется текст. Выходные данные дополнительно уточняются путем сопоставления со словарем из нашей базы данных с использованием алгоритма сопоставления ближайшего соседа. Наконец, мы используем эти выходные данные в качестве ключевых слов для поиска в нашей базе данных. Во время поиска веса tf-idf (частота термина - обратная частота документа) используются для ранжирования возвращаемых результатов. Самый высокий результат, если он существует, объявляется как наш прогноз для каждого изображения обложек книги. Мы создали нашу поисковую систему на основе Apache Solr, так что наша система хорошо масштабируется для больших коллекций книг.

Как и у Chen и др. (2010) и Tsai и др. (2011), мы сообщаем о точности и запоминании при запросе 454 изображений. Точность определяется как доля правильно идентифицированных книг к заявленным правильным книгам, в то время как отзыв определяется как доля правильно идентифицированных книг ко всем запросам. Далее мы определили отзыв в topk, который измеряет количество правильно идентифицированных книг, которые появляются в топ-k результатах поиска. Поскольку только один корешок книги в нашей базе данных может быть правильным для каждого запроса, точность на вершине k не имеет значения, поскольку она будет линейно уменьшаться по отношению к K. Мы также оцениваем наши результаты с помощью Reciprocal Rank (RR), который определяется как: RR = 1 / K, где K - позиция в рейтинге первого релевантного документа (в нашем случае - обложек целевой книги). Сообщается средний взаимный рейтинг (MRR) по нескольким запросам. Все эти меры широко используются информационно-поисковым сообществом.



Рис. 5: Отзыв наверху-k во время поиска. Q.Reg означает, что мы используем распознанный текст в качестве ключевых слов во время поиска, в то время как Q.Gt означает, что мы используем наземные заголовки (без метаинформации) в качестве ключевых слов.

На рис. 5 показан отзыв на разных позициях k. Наша модель достигает 96,4% запоминаемости в 5 лучших результатах поиска, демонстрируя эффективность нашей системы обнаружения текста.

При дальнейшем изучении случаев неудач мы обнаружили, что большая часть ошибочных прогнозов была вызвана менее разборчивыми ключевыми словами, используемыми во время поиска. Несколько книг могут иметь очень похожие или даже идентичные заголовки с небольшими различиями в метаинформации, такой как имя автора, имя издателя, номер тома и т.д. Более того, некоторая метаинформация на изображениях обложек книг имеет тенденцию быть размытой и иметь небольшой размер, что затрудняет обнаружение. В таких случаях использование текстового поиска может быть неадекватным. Чтобы лучше понять, как эти факторы могут снизить эффективность нашей модели, мы оцениваем эффективность поиска, запрашивая достоверные названия на изображениях обложек книг. Эти результаты дают разумное представление о потенциальной верхней границе производительности текстового метода (результаты показаны на рисунке 5). Хотя поиск по изображениям может решить эту проблему, он требует больших затрат на хранение и передачу изображений. Остальные случаи сбоев в основном связаны с несовершенными или даже неправильными ограничивающими рамками локализации текста.

Из вышеизложенного сделаем следующие выводы. Предлагаемая система использует современную глубокую нейронную архитектуру для обнаружения и распознавания текста обложки книг с целью идентификации конкретных книг в библиотеке и эффективного управления библиотечным ресурсам. При этом достигается высочайшего уровня производительности для задачи распознавания текста обложки книг на нескольких тестовых наборах данных, сокращая время обучения. Эксперимент по поиску информации проводится с использованием большой базы данных библиотеки для оценки производительности всей системы.

Литература

1. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227.
2. Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Deep features for text spotting. In Computer Vision–ECCV 2014. Springer. 512–528.
3. Lee, D.-J.; Chang, Y.; Archibald, J. K.; and Pitzak, C. 2008. Matching book-spine images for library shelfreading process automation. In Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on, 738–743. IEEE.
4. Lee, C.-Y.; Bhardwaj, A.; Di, W.; Jagadeesh, V.; and Piramuthu, R. 2014. Region-based discriminative feature pooling for scene text recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4050–4057.
5. Neumann, L., and Matas, J. 2012. Real-time scene text localization and recognition. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 3538–3545. IEEE.
6. Nevetha, M., and Baskar, A. 2015. Automatic book spine extraction and recognition for library inventory management. In Proceedings of the Third International Symposium on Women in Computing and Informatics, 44–48. ACM.
7. Quoc, N.-H., and Choi, W.-H. 2009. A framework for recognition books on bookshelves. In Emerging Intelligent Computing Technology and Applications. Springer. 386–395.
8. Shi, B.; Bai, X.; and Yao, C. 2015. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. CoRR abs/1507.05717.
9. Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. Smith, R. 2007. An overview of the tesseract ocr engine. In icdar, 629–633. IEEE.
10. Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, 3104–3112.
11. Taira, P.; Uchida, S.; and Sakoe, H. 2003. Book boundary detection from bookshelf image based on model fitting. In International Symposium on Information Science and Electrical Engineering, 601–604. Citeseer.
12. Tsai, S. S.; Chen, D.; Chen, H.; Hsu, C.-H.; Kim, K.-H.; Singh, J. P.; and Girod, B. 2011. Combining image and text features: a hybrid approach to mobile book spine recognition. In Proceedings of the 19th ACM international conference on Multimedia, 1029–1032. ACM.
13. Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In Computer Vision (ICCV), 2011 IEEE International Conference on, 1457–1464. IEEE.
14. Wang, T.; Wu, D. J.; Coates, A.; and Ng, A. Y. 2012. End-toend text recognition with convolutional neural networks. In Pattern Recognition (ICPR), 2012 21st International Conference on, 3304–3308. IEEE.
15. Yao, C.; Bai, X.; Shi, B.; and Liu, W. 2014. Strokelets: A learned multi-scale representation for scene text recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4042–4049.
16. Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. arXiv preprint arxiv:1212.5701.